

ОКП 42 2953



КОНТРОЛЕРЫ ПРОГРАММИРУЕМЫЕ
ГСП МикроДТ МБ 57.0

Инструкция по эксплуатации

Часть II

Описание языка

ГШ2.598.148 ИЭ1



I. ВВЕДЕНИЕ

I.1. Общие сведения

Настоящий документ содержит сведения, необходимые для правильного использования контроллеров программируемых ГСП МикроДат МБ 57-0.

Подробно описывается система инструкций языка символического кодирования программируемых контроллеров (ПК), форматы представления данных и правила составления программ.

Кроме описания языка символического кодирования ПК документ содержит примеры по составлению различных программ.

I.2. Программируемые контроллеры

ПК — это электронная машина, доступная для программирования, предназначенная для управления последовательными логическими процессами в условиях промышленной среды в реальном масштабе времени. Структура ПК приведена на рис. III.

ПК выполняет три функции:

- получение информации со входов;
- принятие решения;
- выдачу сигналов на выходы.

Информация представлена в виде дискретных, аналоговых, импульсных сигналов, поступающих от различного рода датчиков, о состоянии оборудования, передающих между собой, являющихся источником информации органов ручного управления.

Обработка информации и принятие решения производится периферийным блоком обработки информации ПК — модулем микропроцессорным МС 59-06 или МС 59-07 (в дальнейшем — процессор).

Он постоянно сравнивает входные сигналы, поступающие от входных устройств, и производит логические операции с этими сигналами по программе, заданной пользователем.

Действия ПК состоят во включении и выключении выходов, управлении внешними устройствами, в качестве которых могут быть применены пускатели, устройства сигнализации и индикации, логические схемы.

Автоматизация объекта управления (машины, группы машин или любого технологического оборудования вообще) заключается в обеспечении его работы с помощью устройства управления. Созданная таким образом автоматизированная система функционирует с учетом параметров состояния объекта управления. Оператор осуществляет

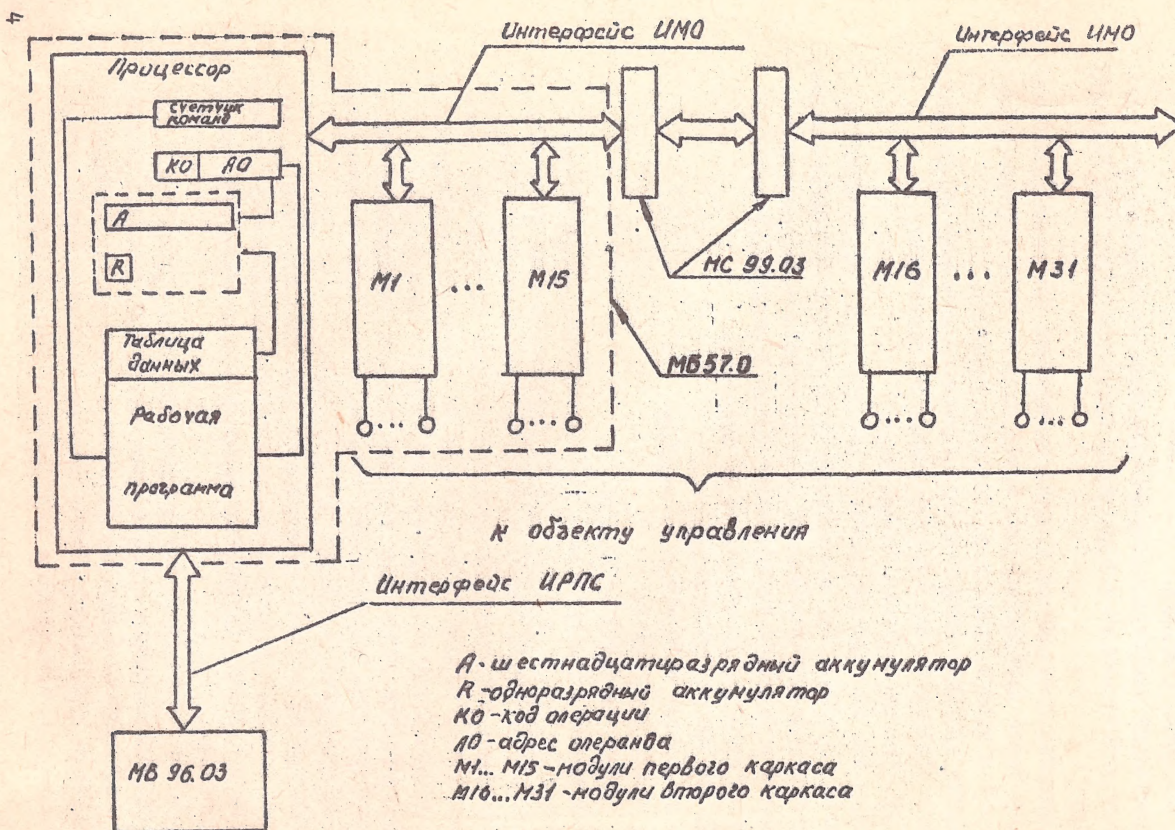


Рис. 1.1

СОДЕРЖАНИЕ

Лист	
3	1. Введение
3	1.1. Общие сведения
3	1.2. Программирование контроллеров
5	1.3. Основные логические схемы
7	2. Описание языка
7	2.1. Общие сведения
10	2.2. Структура памяти ПК
10	2.2.1. Таблица данных
19	2.2.2. Область задания маски обмена
19	2.2.3. Область задания конфигурации
19	2.2.4. Область задания текущего состояния
19	2.2.5. Область диагностики отказов и сбоя
22	2.3. Структура программы
30	2.4. Представление числа
30	2.5. Инструкции логические
38	2.6. Инструкции структурирования
42	2.7. Инструкции для работы с подпрограммами
43	2.8. Инструкции таймеров и счетчиков
53	2.9. Инструкции для работы с данными
59	2.10. Инструкции сравнения
61	2.11. Файловые инструкции
66	2.12. Инструкции экстренного обращения к модулям ввода-вывода
67	2.13. Арифметические инструкции
69	2.14. Примеры по составлению рабочих программ
80	Приложение. Список инструкций языка символьного кодирования

лишь общий контроль за ходом процесса, наблюдает за работой машин и при необходимости берет на себя управление работой всей системы или ее части.

Алгоритм управления задается ПК в виде программы, которую легко заменить или модифицировать.

1.3. Основные логические схемы

Логической схемой называют такую схему, которая выполняет логические операции над одной или более логическими переменными. Логическая переменная — это электрический сигнал, принимающий два различных значения, известные как состояния ИСТИНА и ЛОЖЬ.

Например, логическая переменная А может быть представлена состоянием однополярного переключателя, который может быть либо разомкнут, либо замкнут; следовательно, А принимает два значения. Значение ИСТИНА переменной А соответствует замкнутому состоянию переключателя, значение ЛОЖЬ — разомкнутое (рис. 1.2). Иногда истинное и ложное состояния обозначают соответственно терминами "высокий — низкий уровни", "да — нет", "включено-выключено", "логическая 1 — логический 0".

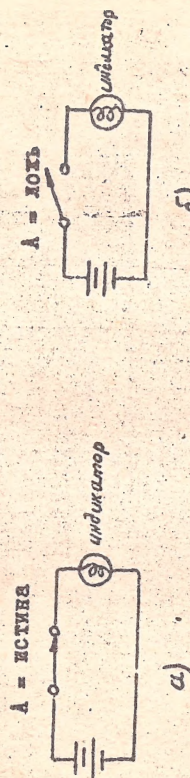


Рис. 1.2

Ниже рассмотрены схемы, реализующие операции двоячной логики.

СХЕМА И

Схема И выполняет операцию логического умножения (конъюнкции) двух или более логических (булевых) переменных. Функция И для двух переменных А и В может быть записана в следующем виде:

$$Y = A \cdot B = A \wedge B \quad (1.1)$$

Можно сказать, что функция И для двух и более логических переменных истинна тогда и только тогда, когда все логические переменные являются истинными. Для случая двух переменных (1.1) значение Y является истинным, когда обе переменные А и В истинны.

Комбинации переменных A, B и соответствующие им значения функции можно представить в виде таблицы истинности (табл. 1)

Значения переменных		Значение функции
A	B	$Y = A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Таблица 1

СХЕМА ИЛИ
Схема ИЛИ - электронная схема, выполняющая операцию логического сложения (адзундци) двух или более логических переменных. Операция ИЛИ для двух переменных A и B записывается следующим образом:

$$Y = A + B = A \vee B \quad (1.2)$$

На рис. 1.3 изображена электрическая цепь с параллельно соединенными переключателями, которая выполняет операцию ИЛИ.



Рис. 1.3

Индикаторная лампа светится, если замкнут один из переключателей A и B или оба.

Таблица истинности (табл. 2) отображает операцию ИЛИ.

Значения переменных		Значение функции
A	B	$Y = A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Таблица 2

Продолжение

Инструкция	Мnemonic-код	Интервал адресов в восьмеричном формате	Наименование	Время выполнения, мкс	
				при выполнении	при игнорировании
Арифметическая	САС	0 - 377	Сложить арифметически слово	200	5
	ВАС	0 - 377	Вычесть арифметически слово	200	5
	ПРС	0 - 7777	Параметр слова	5	5
	ПРК	0 - 4095 ₁₀	Параметр константы в десятичном формате	5	5
	НОП	без адреса	Нет операции (только выходная инструкция)	120	5

Инструкция	Имя-код	Интервал адресов в восьмеричном формате	Наименование	Время выполнения, мкс	при выполнении	при инициализации
------------	---------	---	--------------	-----------------------	----------------	-------------------

Для работы с данными	ЧВС	ПВБ	0 - 377	Обменять содержимым бит старший и младший	120	5
	СВС	СРС		Сдвинуть битовый сдвиг к старшим битам	120	5
Сравнение	СЛМ	СЛМ		Сдвинуть битовый сдвиг к младшим битам	120	5
	СЛМ	СЛМ		Сравнить на "меньше"	120	5
	СЛМ	СЛМ	0 - 377	Сравнить на "больше"	120	5
	СЛМ	СЛМ		Сравнить на "равно"	120	5
Файловая	ПФБ	ПФБ	0 - 377	Переместить файл	перемешан	5
	ЧФБ	ЧФБ	0 - 377	Записать слово в файл	300	5
	ПФМ	ПФМ	0 - 177	Читать слово из файла	300	5
	ПФМ	ПФМ	0 - 177	Переместить файл в модуль	перемешан	5
Экстренного обращения и модуль ввода	ЧТР	ЧТР	0 - 177	Переместить из модуля в файл	перемешан	5
	ЗТР	ЗТР	0 - 177	Читать из регистра	200	5
	ЗТР	ЗТР	0 - 177	Записать в регистр	200	5
	ЗТР	ЗТР	0 - 177	Читать из регистра	200	5

Схема НЕ (инвертор)

Инвертор - это электронная схема, выполняющая операцию НЕ над логической переменной.

Операцию НЕ для переменной А можно записать следующим образом:

$$N = \bar{A}$$

Таблица истинности (табл. 3) отображает операцию НЕ.

Таблица 3

A	Y
0	1
1	0

2. ОПИСАНИЕ ЯЗЫКА

2.1. Общие сведения

В контроллерах для запоминания инструкций, выполняющих логические операции, временную задержку, арифметическую обработку и т.д., служит программируемая память.

Совокупность необходимых инструкций или программа вводится в программируемую память пользователя при помощи прибора программирования и отладки МВ 96.03 (в дальнейшем - МВ96.03). Порядок работы МВ 96.03 описан в инструкции по эксплуатации МВ96.048.023 МЗ.

Исходное описание алгоритма управления объектом должно быть представлено на языке релейно-контактных схем или на языке алгебры Буля.

МК сообщается с объектом управления с помощью модулей ввода-вывода, которые установлены в картах ввода-вывода.

Вся информация, обрабатываемая процессором, сосредоточена в таблице данных (ТД).

В процессе работы МК инструкции программы последовательно, начиная с первой, считываются процессором из памяти рабочей программы (ПРП) и выполняются, т.е. изменяется состояние ТД.

После выполнения последней инструкции происходит обмен информацией между определенной областью ТД с одной стороны и модулем

вывода с другой. Эти две фазы работы ПК - "счет" и "обмен" непрерывно циклически повторяются. Циклы работы процессора приведены на рис. 2.1. Первый цикл, после начального зауста, процессор проходит с результатом равным нулю. Это значит, что при первом проходе, независимо от результата вычисления функции, выхода, задерживаемые во всей программе, обнаружены и сбрасываются снова, на которых будут запускаться таймеры на включение.

Процессор содержит указатель адреса инструкции, содержание которого увеличивается на 1 при выполнении очередной инструкции. При выполнении последней инструкции программы указатель адреса устанавливается в 0000 (адрес первой инструкции программы).

Процессор содержит два аккумулятора для накопления данных. Один одноразрядный аккумулятор для операций с логическими инструкциями и шестнадцатизрядный для операций с многоразрядными инструкциями.

ПК может работать в трех режимах:

"Работа" (RAB) - выполнение рабочей программы (PI);

"Программирование/Тестирование" (П/Т) - ввод и редактирование PI, формирование конфигурации памяти процессора, ввод констант в ПП, ввод буквенно-цифровых сообщений;

"Работа/Отладка" (P/O) - выполнение PI, останов выполнения цикла PI, блокировка обмена с модулями ввода/вывода, снятие останова выполнения цикла, блокировка обмена с модулями ввода/вывода.

ПК имеет последовательный канал связи с МВ 96.03 и с устройством программирования и отладки (МВ 96.02). Для ввода программы в ПК с помощью МВ 96.03 принята табличная форма записи.

Вводится программа по одной инструкции (строка таблицы). Для МВ 96.02 принят язык релейно-контактных схем. Программа вводится по одной цепи релейно-контактной схемы.

Продолжение

Инструкция	Мнемо-код	Интервал адресов в восьмеричном формате	Наименование	Время выполнения, мкс	
				при выполнении	при игнорировании
Для работы с подпрограммами	ВПП	0 - 7	Вызвать подпрограмму	120	5
	НПП	0 - 7	Начало подпрограммы	120	5
	НПП	77	Конец массива подпрограммы	120	120
Таймеров и счетчиков	ТОД	0 - 377	Таймер на отключение (0,1 с)	250	5
	ТОС		Таймер на отключение (1 с)	250	5
	ТВД		Таймер на включение (0,1 с)	250	5
	ТВС		Таймер на включение (1 с)	250	5
	ТНД		Таймер накапливающий (0,1 с)	250	5
	ТНС		Таймер накапливающий (1 с)	250	5
	СЧВ		Счетчик на вычитание	250	5
	СЧО		Счетчик на сложение	250	5
	СЧЦ		Счетчик циклов	250	5
Для работы с данными	ЧТС	0 - 377	Читать слово	120	5
	ЗПС		Записать слово	120	5
	ПУС		Поразрядно умножить слово	120	5
	ПСС		Поразрядно сравнить слово	120	5
	ЗБМ		Записать байт младший	120	5
	ЗБС		Записать байт старший	120	5
	ЧБМ		Читать байт младший	120	5

СПИСОК ИНСТРУКЦИЙ ЯЗЫКА СИМВОЛИЧЕСКОГО КОДИРОВАНИЯ

Инструкция	Имя-код	Имя-код в восьмеричном формате	Наименование	Время выполнения, мкс	при выполнении, мкс	при выполнении, мкс
Логическая	-11-	0 - 37717	Прямой опрос входа	7		5
	-И-	0 - 37717	Инверсный опрос входа	7		5
	-(-)	0 - 37717	Включить выход	9		5
	-(S)-	0 - 37717	Включить выход и зафиксировать	9		5
	-(R)-	0 - 37717	Включить выход и зафиксировать	9		5
	Т	без адреса	Начало одного ветвления	7		5
	Т	То же	Конец одного ветвления	7		5
	Т	То же	Перенос маркера в начало ветвления	7		5
	НСТ	0 - 7	Начало сегмента	120		120
	3СТ	0 - 7	Закрывать сегмент	120		120
	ОСТ	0 - 7	Открывать сегмент	120		120
Структурно-вспомогательная	НСТ	77	Сбросить сегмент	120		5
	НСТ	77	Конечный цикл программы	120		5
	3СТ	77	Программировать оставшиеся	4000		5
	НЕМ	0 - 76	Начало блока	120		5
	НЕМ	77	Возврат из подпрограммы	120		5
	ПЕМ	0 - 76	Перейти на блок	120		5
	МПС	0 - 17	Установить номер сектора ТМ	120		5

Цикл работы процессора

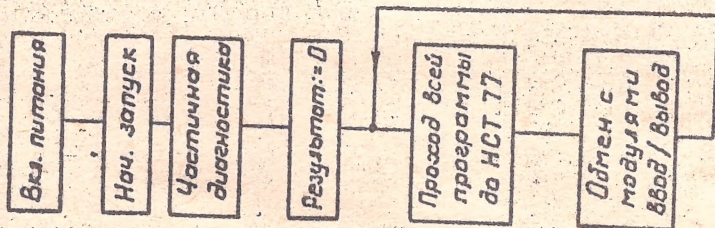


Рис. 2.1

2.2. Структура памяти ПК

Вся программная память MS 59.06 имеет объем 2000 слов, а MS 59.07 имеет объем памяти 4000 слов. Программируемая память состоит из ТД, защищенной области, области задания текущего состояния, области задания конфигурации, области задания текущего состояния, области адресности отсчетов и сбросов. Структура памяти приведена на рис. 2.2.

Объем ТД и ПРП задается пользователем в процессе программирования: Объем ТД и ПРП не может превышать для MS 59.06 - 37178 слов, а для MS 59.07 - 77178 слов. Программируемая память в общем случае может состоять из 8 секторов (00,01 ... 07) - для MS 59.06 и из 16 секторов (00,01 ... 17) - для MS 59.07.

Объем одного сектора составляет 4008 слов (рис. 2.3). В пределах одного сектора адреса слов от 000 до 377g.

Процессор может читать информацию из ТД по четырехзначному адресу по трехзначному адресу в восьмеричном формате согласно списку инструкций языка ПК, приведенному в приложении.

2.2.1. Таблица данных

Минимальной информационной единицей, обрабатываемой процессором, является бит. Бит принимает два возможных логических значения "0" либо "1". Канал модуля ввода или вывода дискретных сигналов имеет отображение в ТД в виде бита. Формирование адреса бита приведено на рис. 2.4.

Упорядоченная совокупность из шестнадцати бит образует слово.

Биты в слове нумеруются двузначными восьмеричными числами 00, 01 ... 17. Каждые восемь бит слова образуют нумером, а старшие - первый байт слова.

Обработанные программой потребителя данные (слова и биты) распределяются в ТД.

В ТД находится информация о состоянии модулей ввода-вывода. В фазе обмена процессор производит обмен информацией между дискретными модулями ввода-вывода и ТД. Схема обмена между ТД и модулями ввода-вывода приведена на рис. 2.5.

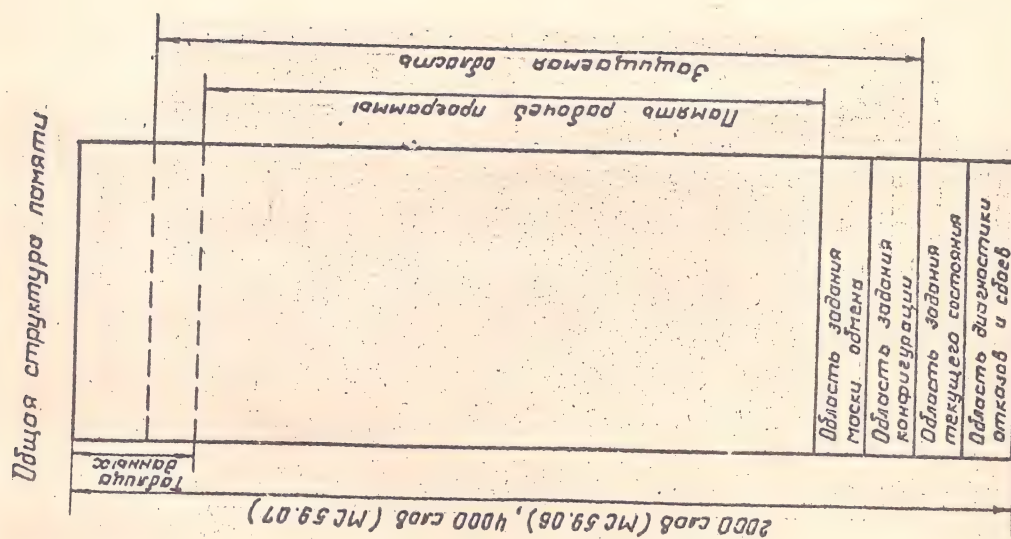
В ТД в слове, свободных от обмена между модулями ввода-вывода и ТД, могут быть организованы таймеры, счетчики, в свободные слова можно записывать биты промежуточной информации, различные константы. Примеры возможного распределения памяти, таблицы для распределения модулей ввода-вывода, побитное распределение ТД, побитное распределение приведены на рис. 2.6, 2.7, 2.8, 2.9.

Продолжение

Нумерация инструкций	Минимум	Полное значение	Нумерация инструкций	Минимум	Полное значение
0074	-1- 05000		0109	-1- 05000	
0075	-1- 05001		0110	-1- 05001	
0076	-1- 05002		0111	-1- 05002	
0077	-1- 05003		0112	-1- 05003	
0078	-1- 05106		0113	-1- 05115	
0079	-1- 05000		0114	-1- 05000	
0080	-1- 05001		0115	-1- 05001	
0081	-1- 05002		0116	-1- 05002	
0082	-1- 05003		0117	-1- 05003	
0083	-1- 05107		0118	-1- 05116	
0084	-1- 05000		0119	-1- 05000	
0085	-1- 05001		0120	-1- 05001	
0086	-1- 05002		0121	-1- 05002	
0087	-1- 05003		0122	-1- 05003	
0088	-1- 05110		0123	-1- 05117	
0089	-1- 05000		0124	НБА 77	
0090	-1- 05001		0125	НПН 77	
0091	-1- 05002				
0092	-1- 05003				
0093	-1- 05111				
0094	-1- 05000				
0095	-1- 05001				
0096	-1- 05002				
0097	-1- 05003				
0098	-1- 05112				
0099	-1- 05000				
0100	-1- 05001				
0101	-1- 05002				
0102	-1- 05003				
0103	-1- 05113				
0104	-1- 05000				
0105	-1- 05001				
0106	-1- 05002				
0107	-1- 05003				
0108	-1- 05114				

二

二



сборная	
Сектор 0	0000
Сектор 1	0372
Сектор 7	0400
Сектор 10	4377
Сектор 11	4400
Сектор 17	7777
МС 59.06	
МС 59.07	

Рис. 2.3

Прозрачность

Номер интер- вал	Минимум	Примечание	Номер интер- вал	Минимум	Примечание
0000	НЕТ 00		0037	НЕТ 00	
0001	НЕТ 00		0038	НЕТ 00	
0002	(-) 05004		0039	НЕТ 00	
0003	(-) 10300		0040	НЕТ 00	
0004	(-) 05000		0041	НЕТ 00	
0005	(-) 10301		0042	НЕТ 00	
0006	(-) 05001		0043	НЕТ 00	
0007	(-) 10302		0044	НЕТ 00	
0008	(-) 05002		0045	НЕТ 00	
0009	(-) 10303		0046	НЕТ 00	
0010	(-) 05003		0047	НЕТ 00	
0011	(-) 05004		0048	НЕТ 00	
0012	НЕТ 00		0049	НЕТ 00	
0013	(-) 05004		0050	НЕТ 00	
0014	НЕТ 00		0051	НЕТ 00	
0015	НЕТ 00		0052	НЕТ 00	
0016	(-) 10304		0053	НЕТ 00	
0017	(-) 05000		0054	НЕТ 00	
0018	(-) 10305		0055	НЕТ 00	
0019	(-) 05001		0056	НЕТ 00	
0020	(-) 10306		0057	НЕТ 00	
0021	(-) 05002		0058	НЕТ 00	
0022	(-) 10307		0059	НЕТ 00	
0023	(-) 05003		0060	НЕТ 00	
0024	(-) 05004		0061	НЕТ 00	
0025	НЕТ 00		0062	НЕТ 00	
0026	(-) 05004		0063	НЕТ 00	
0027	НЕТ 00		0064	НЕТ 00	
0028	НЕТ 00		0065	НЕТ 00	
0029	(-) 10310		0066	НЕТ 00	
0030	(-) 05000		0067	НЕТ 00	
0031	(-) 10311		0068	НЕТ 00	
0032	(-) 05001		0069	НЕТ 00	
0033	(-) 10312		0070	НЕТ 00	
0034	(-) 05002		0071	НЕТ 00	
0035	(-) 10313		0072	НЕТ 00	
0036	(-) 05003		0073	НЕТ 00	

Пример реализации дешифратора на четыре
 входа и шестнадцать выходов
 Входная информация : 05000 - 05003
 Выходная информация : 05100 - 05117.

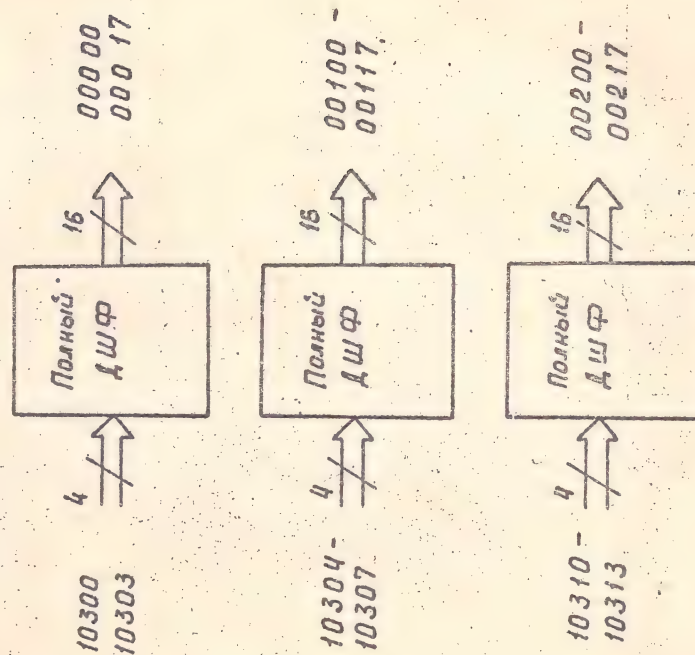


Рис. 2.4

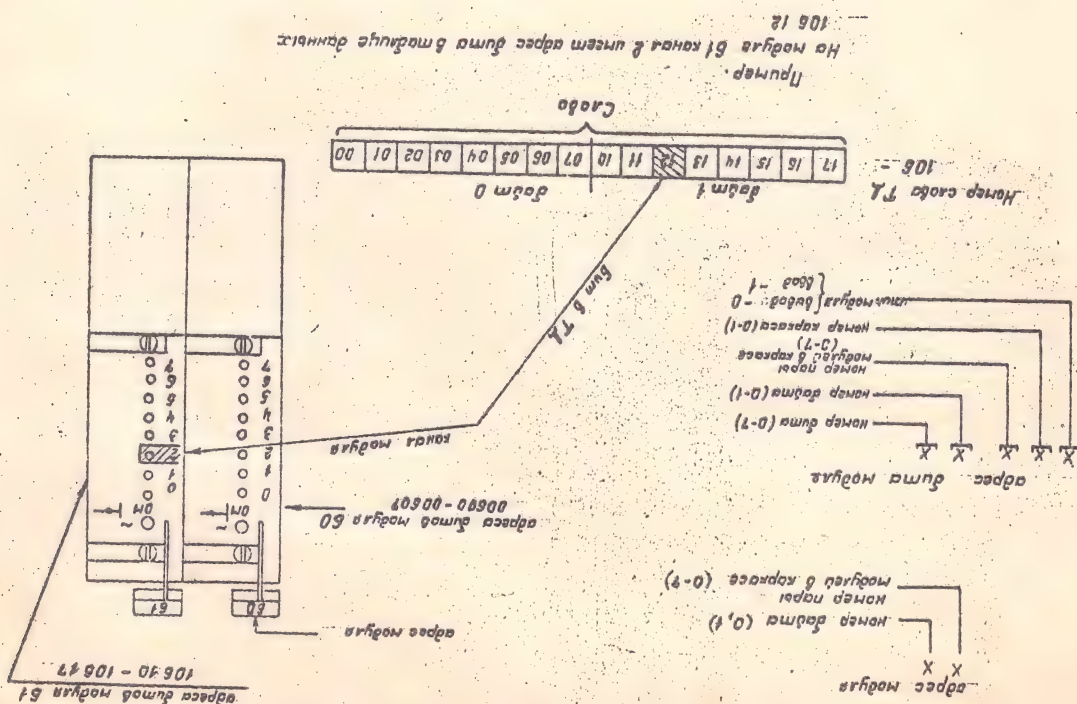
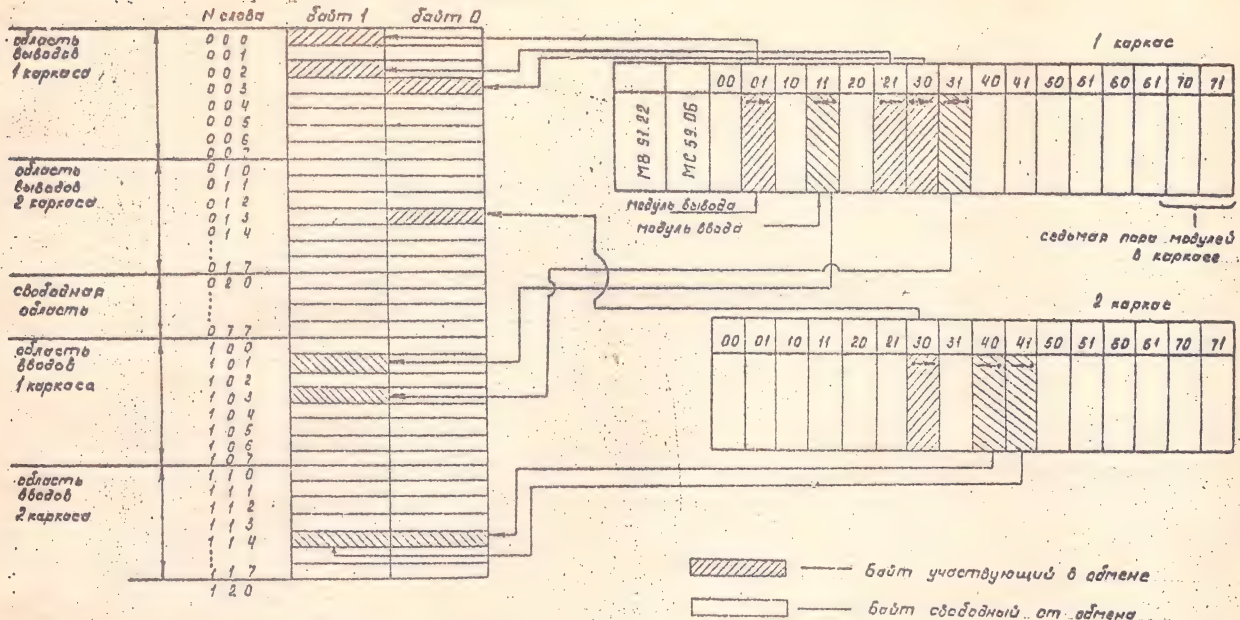


Таблица данных

14



Н слова ТД

х х х

номер пары модулей в каркасе (0-7)

номер каркаса (0-первый каркас 1-второй каркас)

0 - область выводов, 1 - обл. ввода

Пример

0 0 слова 000

каркас 1

модуль ввода

0 1 2 слова 002

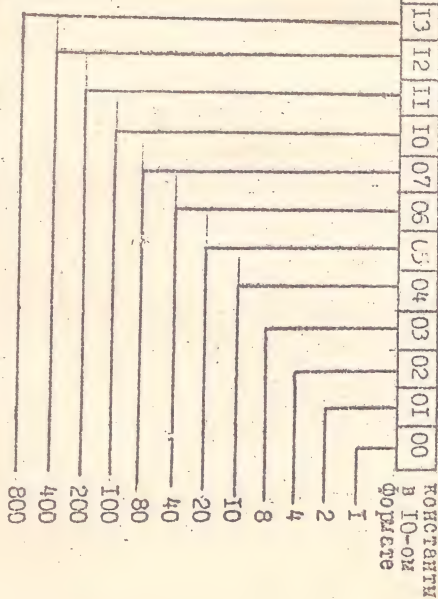
каркас 2

модуль вывода

Рис. 2.5

Программа

Номер инструк- ции	Мнемоник	Приме- чание	Номер инструк- ции	Мнемоник	Приме- чание
0000	НСТ 00				
0001	НМ 00				
0002	СКС 220				
0003	ЧТС 220				
0004	-11- 020 00				
0005	САС 200				
0006	-11- 020 01				



Х - условное обозначение дискретного входа

У - условное обозначение дискретного выхода

Программа перевода двоично-десятичного числа в двоичный код.

Исходные данные: в слове 020 хранится двоично-десятичное число (три горады), в словах 200 - 213 - весовые константы, в слове - 220 - результат (двоичный код).

СКС 050

-11- 13

-11- 11

-11- 12

НМ 00

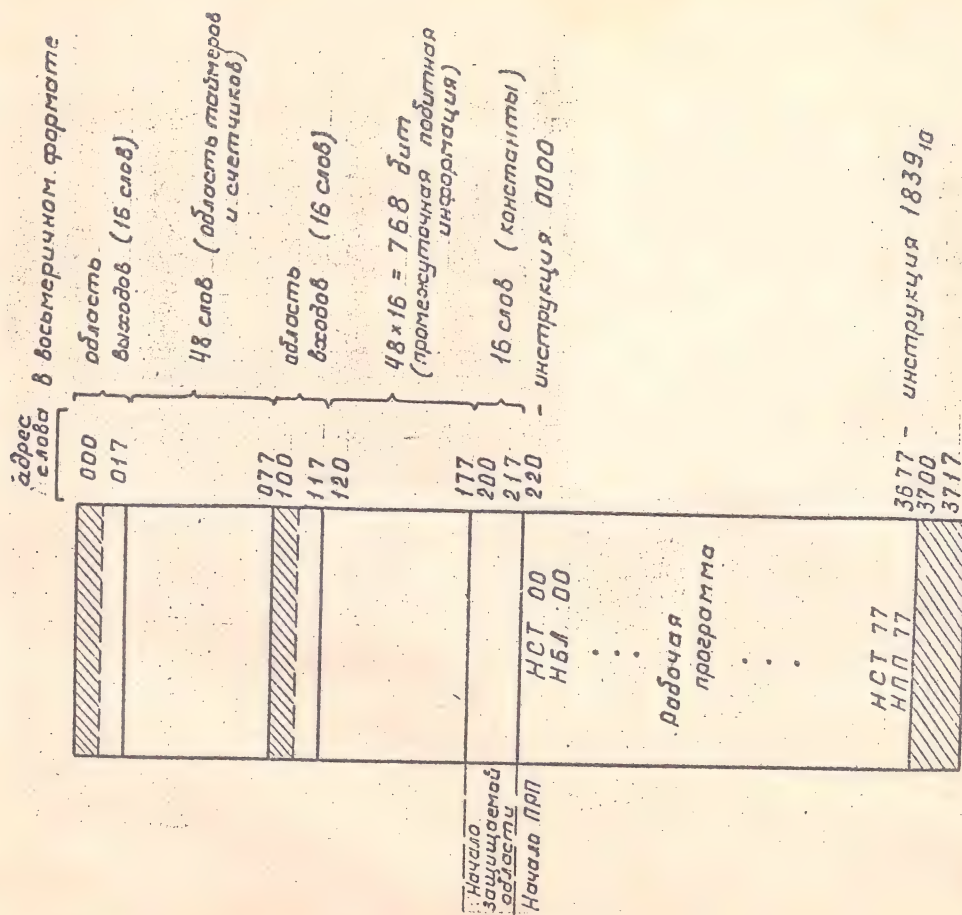
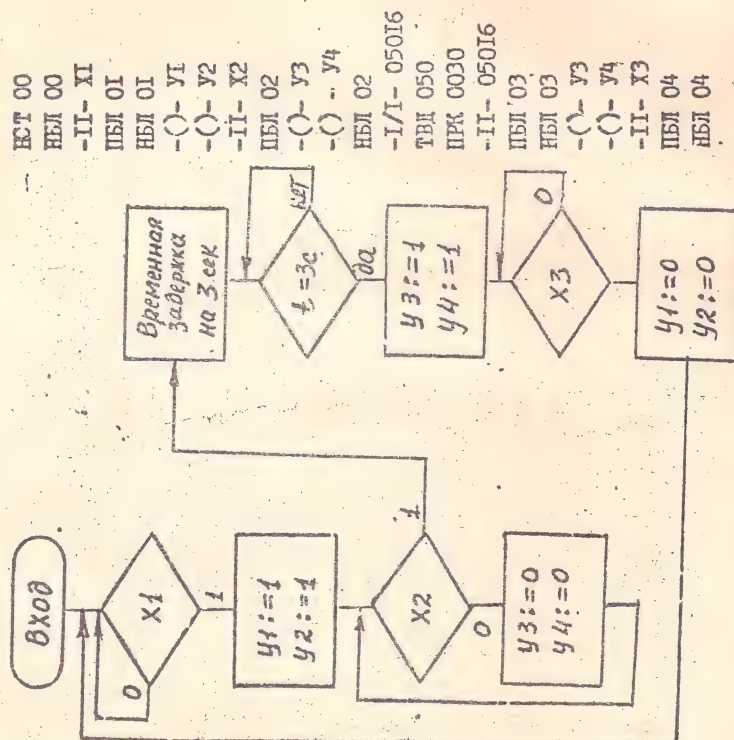
НСТ 77

НМ 77

Номер инструк- ции	Именовод	Приме- чание	Номер инструк- ции	Именовод	Приме- чание
0007	-() - 00000		0023	НСТ 77	
0008	ЧСФ 050		0024	НПП 77	
0009	ПРС А				
0010	ПРС W1				
0011	ПРК N				
0012	ЧСФ 051				
0013	ПРС В				
0014	ПРС W2				
0015	ПРК N				

Пример реализации алгоритма, представленного блок-схемой.

Алгоритм



Рабочая программа, в которой подбитной обработки - 1800, таймеров - 40, выполняется - за 26 мс.

Расчет:

1800 x 7 мкс (время выполнения подбитной инструкции) = 12 мс
40 x 250 мкс (время выполнения инструкции таймеров) = 10 мс
Обмен = 4 мс.

Рис. 2.6

Программируемый контроллер Распределение модулей Ввода - Вывода

Модуль	Бит	Обозначение сигнала	Элементы схемы	Номер провода	Примечание
Служб.	0				
	1				
	2				
	3				
	4				
	5				
	6				
	7				

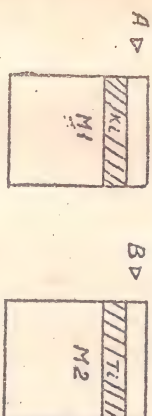
Модуль	Бит	Обозначение сигнала	Элементы схемы	Номер провода	Примечание
Служб.	0				
	1				
	2				
	3				
	4				
	5				
	6				
	7				

Рис. 2.7

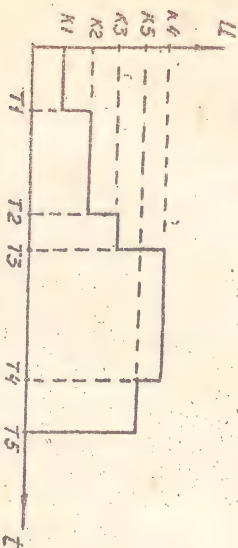
Реализация временной функции.

$$U = f(t),$$

Исходная информация: массив M_1 - набор констант K_1 и M_2 - констант T_1 .



A - начальный адрес массива M_1 ,
 B - начальный адрес массива M_2
 W_1, W_2 - адреса слов, N - число констант в массиве.
График функции



Программа

Номер инструк-ции	Мнемоник	Примечание	Номер инструк-ции	Мнемоник	Примечание
0000	НСТ 00		0016	ПЫ 02	
0001	ВЫ 00		0017	ВЫ 02	
0002	СВС 050		0018	ТВЫ 052	
0003	СВС 051		0019	ПРС W/2	
0004	ПЫ 01		0020	-11-05216	
0005	ВЫ 01		0021	СВС 052	
0006	-1/1-000 00		0022	ПЫ 01	

Программа

Номер инструкции	Мнемоник	Примечание	Номер инструкции	Мнемоник	Примечание
0000	НСТ 00		0036	НСТ 02	ручной режим
0001	НБЛ 00		0037	НБЛ 00	
0002	-II- IOIOI	пуск авт. реж.	0038	-I/I- IOI 03	
0003	-I/I- IOIOO		0039	-II- IOI 06	A - вперед
0004	ОСТ 01		0040	-(-)- 00002	
0005	ЗСТ 02		0041	-I/I- IOIO2	
0006	НБЛ 01		0042	-II- IOIO7	A - назад
0007	-II- IOIOI		0043	-(-)- 00003	
0008	-II- IOIOO		0044	-I/I- IOIO5	
0009	ОСТ 02		0045	-II- IOIOO	B - вперед
0010	ЗСТ 01		0046	-(-)- 00005	
0011	НБЛ 01		0047	-I/I- IOIO4	
0012	НБЛ 01		0048	-II- IOIOI	B - назад
0013	-I/I- IOIOI		0049	-(-)- 00006	
0014	НБЛ 00		0050	НСТ 77	
0015	НСТ 01	авт. реж.	0051	НПН 77	
0016	НБЛ 00				
0017	-I/I- IOIO2	Призвездное В исходное состояние			
0018	-(-)- 000 03				
0019	-I/I- IOI 04				
0020	-(-)- 00006				
0021	-II- IOI 02				
0022	-II- IOIO4				
0023	НБЛ 01	движения А вперед			
0024	НБЛ 01				
0025	-I/I- IOIO3				
0026	-(-)- 00002				
0027	-II- IOIO3				
0028	НБЛ 02				
0029	НБЛ 02	A-назад B-вперед			
0030	-I/I- IOIO2				
0031	-(-)- 00003				
0032	-I/I- IOIO5				
0033	-(-)- 00005				
0034	-II- IOIO5				
0035	НБЛ 00				

Программируемый контроллер
Подбитное распределение
таблицы данных

Процессор
Объем 1А
Слова от — до —

Слово	Бит	Описание	Слово	Бит	Описание
0	0			0	0
0	1			0	1
0	2			0	2
0	3			0	3
0	4			0	4
0	5			0	5
0	6			0	6
0	7			0	7
1	0			1	0
1	1			1	1
1	2			1	2
1	3			1	3
1	4			1	4
1	5			1	5
1	6			1	6
1	7			1	7
0	0			0	0
0	1			0	1
0	2			0	2
0	3			0	3
0	4			0	4
0	5			0	5
0	6			0	6
0	7			0	7
1	0			1	0
1	1			1	1
1	2			1	2
1	3			1	3
1	4			1	4
1	5			1	5
1	6			1	6
1	7			1	7

Комментарий

Рис. 2.8

Процессор _____
 Объем ТД _____
 Служба от _____ до _____

Слѣдо	Описание	Вѣст мѣдѣи
0	Вѣст старѣи	
1		
2		
3		
4		
5		
6		
7		
0		
1		
2		
3		
4		
5		
6		
7		
0		
1		
2		
3		
4		
5		
6		
7		
0		
1		
2		
3		
4		
5		
6		
7		

Модуль 10	Вайт	Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Словно 101	0	4 5 6 7	X4 X5 X6 X7			

Модуль II	Вариант	Вит	Обозначение сил-надов	Элементы схемы	Номер провода	Примечание
Слово 101	I	0	XIO			
		1	XII			
		2				
		3				
		4				
		5				
		6				
		7				

Ідентифікація конфігурації ПК.

С помощью МБ 96.03 надо ввести:

НАЧАЛО 30 0120

НАЧАЛО ПРЯМОГО

ВРЕМЯ ЦИКЛА 0377

В/В НАРТАС I

XXV

В/В КАРКАС 2

Предполагается два режима работы: ручной и автоматический. Автоматический режим.

Исходное состояние: тележки А и В находятся над датчиками X2, X4.

Работа: тележка А движется от датчика Х2 к Х3, при подходе к датчику Х3 начинается движение тележки В от Х4 к Х5, а тележка А движется от Х3 к Х2. После подхода к датчику Х5, тележка В движется к датчику Х4.

Каркас III:

	NB	91.22
	MC	59.06
	MC	35.18-01
10		
01		MC 34.08-03
11		MC 34.08-03

Распределение модулей входов

Модуль 00	Байт	Бит	Обозначение сигнала	Элементы схемы	Номер проволки	Примечание
Сторона 000		0	-			
		1	-			
		2	У2			
		3	У3			
	0	4	-			
		5	У5			
		6	У6			
		7	-			

Модуль IO	Бит	Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Словно IOI		0	X0			Автомат. рет. Пуск
		1	X1			
		2	X2			
		3	X3			

2.2.2. Область задания маски облена (рис. 2.10).

Маска обрана - это информация о наличии дискретного модуля в кардесе и разрешение на обмен информацией с этим модулем.

Маску обмена можно задавать, пользуясь специальной функцией

ИВ 96.03.

Соответствие бита слова маски обмена и номера модуля в кар-
тасе приведено на рис. 2.11.

2.2.3. Область заданной конфигурации

Под конфигурацией системы понимается начало защищаемой области, начало рабочей программы и время цикла. Эти три параметра пользователя необходимо ввести с помощью ИВ 96.03.

Часть памяти потребителя, называемая защитимой областью, может храниться, по выбору потребителя, в ОЗУ с подпиткой от резервного источника питания либо в ПЗУ с ультрафиолетовым способом стирания. Содержимое защитимой части памяти может быть введено на магнитную ленту или введено с магнитной ленты. Минимальный адрес, с которого может начинаться защитимая область, равен 120₈ (попер слова₈). С этого же адреса может начинаться и прт.

Для МС 59-06 максимально возможная ПРП занимает объем от 120⁸ до 3677⁸ слов или имеет от 0 до 1903¹⁰ инструкций.
Для МС 59-07 максимально возможная ПРП занимает объем от 120⁸ до 7677⁸ слов или имеет от 0 до 3951¹⁰ инструкций.

максимальное время цикла — 426 мс соответствует коду 03778.
Превращение времени цикла фиксируется в системном отказе (отказ ПВЕ). Таблица соответствия времени цикла — коду, приведенная на рис. 2.12.

2.2.4. Область задания текущего состояния.

Процессор при прохождении рабочей программы используется информационной области задания текущего состояния, и при определенных условиях ее изменяет. Структура области задания текущего состояния приведена на рис. 2-13. При выполнении инструкций структурирования информация этой области изменяется.

2.2.5. Область диагностики отказов и сбоя.

Доступ в область отказов и сбоев дает использование инструкции УНС. На рис. 2.14, рис. 2.15, рис. 2.16 раскрыто содержание слов данной области.

Бит в любом разряде означает отказ, ноль-норму.

Область задания маски обмена

МС59-06 МС59-07

Маска обмена входов 1-го каркаса	3700	(7700)
Маска обмена входов 2-го каркаса	3701	(7701)
Маска обмена входов 1-го каркаса	3702	(7702)
Маска обмена входов 2-го каркаса	3703	(7703)

Область задания конфигурации

Начало защищаемой области	3704	(7704)
Начало ПРП	3705	(7705)
Время цикла	3706	(7706)
Контрольная сумма	3707	(7707)

Область задания текущего состояния

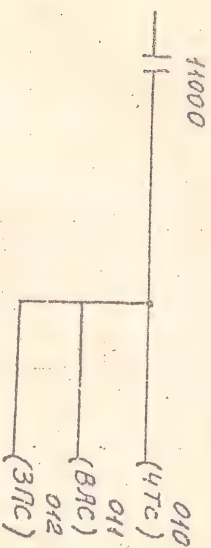
S1	S0	3710	(7710)
S3	S2	3711	(7711)
S5	S4	3712	(7712)
S7	S6	3713	(7713)

Область диагностики отказов и сбоя

Системные отказы	3714	(7714)
и сбоев	3715	(7715)
Отказы 1-го каркаса	3716	(7716)
Отказы 2-го каркаса	3717	(7717)

Рис. 2.10

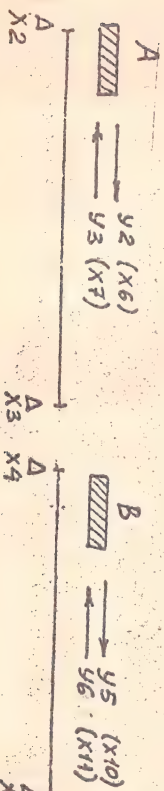
Номер слова	Содержимое в десятичном формате
010	2048
011	0032
012	2016



Программа

Номер инструкции	Мнемоника	Примечание
0000	НСТ 00	
1	НВМ 00	
2	-11- 110 00	
3	ЧРС 010	
4	ВАС 011	
5	ЗПС 012	
6	НСТ 77	
7	НПН 77	

2.14. Примеры по составлению *рабочих* программ. Реализация технологического процесса движения двух тележек



Номер слова	Содержимое в десятичном формате
O10	2048
O11	0032
O12	2080

исходные данные
результат



Программа:

НСТ 00
НЕМ 06
-II- IIO 00
ЧТС 010
САС 011
ЗПС 012
ЕСТ 77
НПН 77

ВЫЧЕСТЬ АРИФМЕТИЧЕСКИ СЛОВАЗ

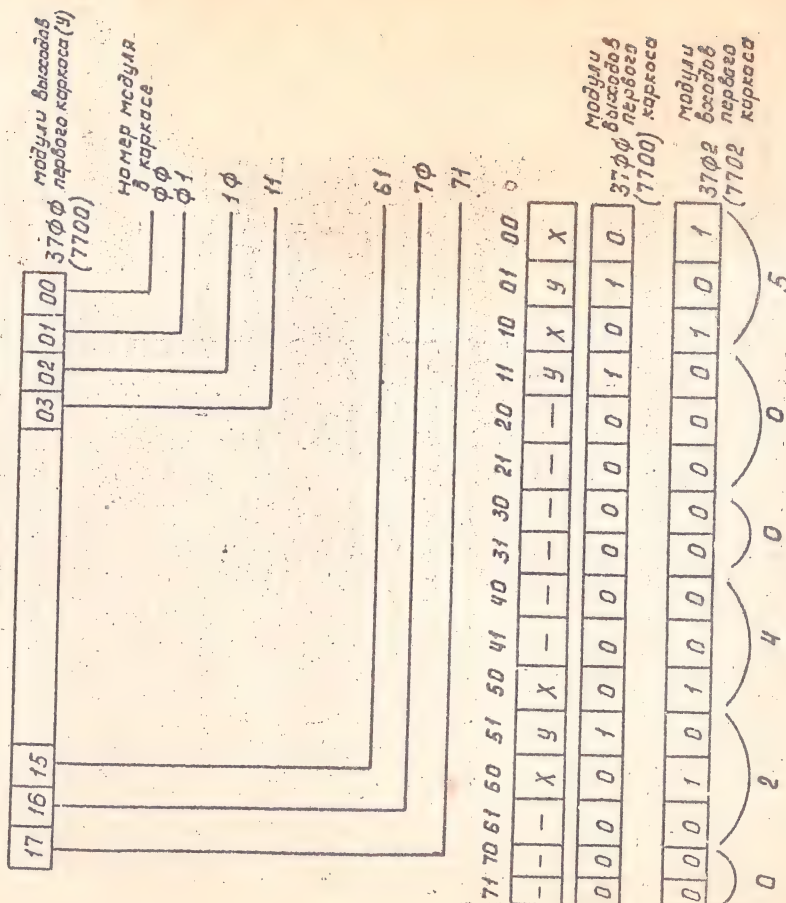
ВАС ППН

Когда цель, в которой находится инструкция вычитания, становится истинной, производится вычитание числа, находящегося в аккумуляторе числа записанного по адресу слова. Если в результате вычитания получено отрицательное число, то включается бит I4.

Пример.

Содержимое слов (O10) и (O11) равно соответственно 2048 и 32. Необходимо вычесть 32 из 2048 и результат записать по адресу (O12).

Соответствие бита слова маски адреса и номера модуля в картесе



Модуль выходов - У,

Модуль входов - Х

В слово 3700 надо записать константу в восьмеричном формате 010012, в 3701 - 000 000, в 3702 - 024 005, в 3703 - 000 000.

Рис. 2.11

Время цикла (ис)	Код в восьмеричном формате
10	0006
30	0022
50	0036
70	0052
90	0366
110	0102
130	0110
150	0123

Рис. 2.12.

2.3. Структура программы

Программа представляет собой упорядоченную последовательность инструкций, т.е. каждая инструкция имеет свой порядковый номер - четырехзначное десятичное число 0000, 0001 ... и т.д.

Инструкция - это распределение, выполняемое процессором.

Процессор выполняет не только инструкции, а только те, которые входят в набор инструкций, заложенных в его память (см. приложение).

Инструкция является наименьшей самостоятельной единицей программы и в общем случае состоит из оператора и операнда.

Инструкция

оператор

операнд

Оператор описывает выполнение процессором функции. Она указывает, что должен делать процессор.

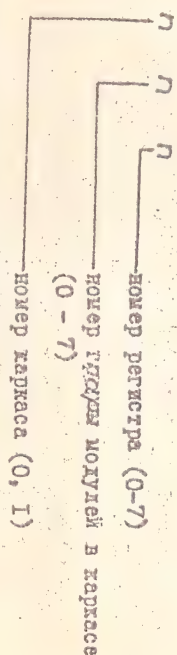
Операнд указывает данные, необходимые для выполнения инструкции.

Структура программы приведена на рис. 2.17.

Программа должна начинаться инструкцией $INT\ 00$ - начало сегмента 0, $INT\ 00$ - начало блока 0 и заканчивается: $INT\ 77$ - конец цикловой программы, $INT\ 77$ - конец несдвига подпрограммы.

Пример бланка для написания программы приведен на рис. 2.18. Программа подразделяется на сегменты, начало каждого из ко-

Структура адреса регистра



Распределение регистров в группе модулей следующее:

номер регистров	номер модулей
0	4
1	5
2	6
3	7

N - номер пары модулей в картесе

2.13. Арифметические инструкции

Эти инструкции дают возможность арифметической обработки данных в подпрограмме. С помощью арифметических инструкций можно обрабатывать целые положительные числа в пределах от 0 до 4095. Бит 14 в служебных разрядах вычислителя, если в результате арифметических операций результат вышел за указанные пределы. Арифметические операции - условные.

СЛОЖИТЬ АРИФМЕТИЧЕСКОЕ СЛОВО

САС nnn

адрес слова

Кодка цепи, в которой находится инструкция сложения становится истинной, производится операция сложения чисел, находящихся в аккумуляторе и по адресу слова, следующего за САС. Если в результате сложения получено число больше чем 4095, то вычислитель бит 14 в слове, где записан результат.

Пример.

Содержимое слов (010) и (011) равно соответственно 2048 и 32. Эти числа необходимо сложить и результат поместить по адресу (012).

ПЕРЕМЕСТИТЬ ФАЙЛ В МОДУЛЬ

ПРМ ХХ

ПРС ХХХХ - начальный адрес памяти модуля

ПРС ХХХХ - начальный адрес ТД

ПРК ХХХХ - длина файла L

ПЕМ Х Х Х

номер регистра

номер группы модулей в картесе

номер картеса

При выполнении этой инструкции файл из ТД переносится в память модуля по указанному адресу за один скан, если записываемое выражение находится в состоянии "1".

ПЕРЕМЕСТИТЬ ИЗ МОДУЛЯ В ФАЙЛ

ПМФ Х Х Х

ПРС Х Х Х Х - начальный адрес памяти модуля

ПРС Х Х Х Х - начальный адрес ТД

ПРК(К) Х Х Х Х - длина файла

При выполнении этой инструкции файл из памяти модуля переносится в ТД по указанному адресу.

Программирование отдельных модулей, входящих в состав ПК, приведено в технических описаниях или в инструкциях по эксплуатации на модули.

2.12. Инструкции экстренного обращения к модулям ввода-вывода

ЧИТАТЬ ИЗ РЕГИСТРА

ЧТР nnn

При выполнении этой инструкции содержимое регистра модуля передается в *аккумулятор*.

ЗАПИСАТЬ В РЕГИСТР

ЭПР nnn

При выполнении этой инструкции содержимое *аккумулятора* ТД передается в регистр модуля.

Инструкции ЧТР и ЭПР применяются при работе с модулями, имеющими регистры.

Область задания текущего состояния

17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
сегмент									01	сегмент						00
3710 (7710)									03	02						
3711 (7711)									05	04						
3712 (7712)									07	06						
3713 (7713)																

7 6 5 4 3 2 1 0

номер активного блока

0 0 - одноступенчатое

0 1 - инициализация

1 1 - вычисление

Рис. 2.13

Системные отказы в слове 3714 (7714)

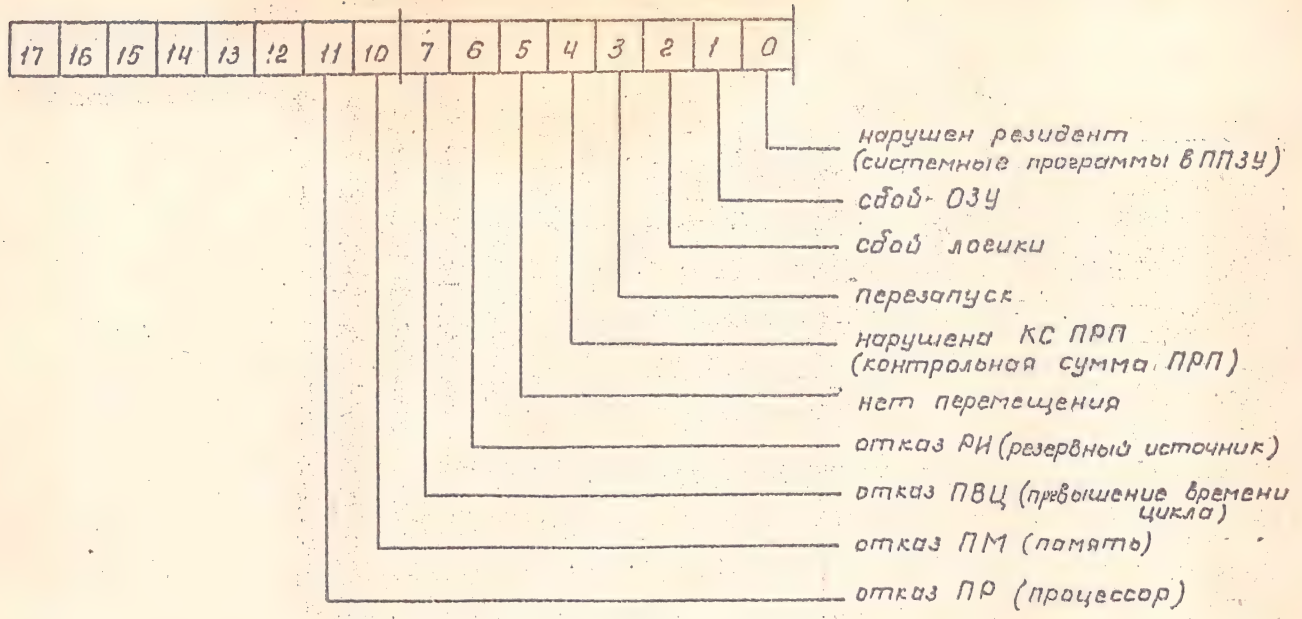


Рис. 2.14

ЗАПИСАТЬ СЛОВО В ФАЙЛ
30F XXX - адрес счетчика
PRS XXXX - начальный адрес файла
PRS XXXX - адрес слова
PRK XXXX - длина файла
При изменении заданного выражения из "0" в "1" содержание по адресу слова переиспользуется в начальный адрес файла. При следующем изменении заданного выражения из "0" в "1" содержание по адресу слова заново заносится в следующий адрес файла, т.е. начальный адрес файла + 1.

Пример.
Записать в ПД константы:

Номер слова	Константы
0036	123 123
0037	177 177

} исходные данные

Программа

Номер инструк-ции	Мнемоника	Примечание	Номер инструк-ции	Мнемоника	Примечание
0001	EST 00		0013	-11- 030 16	
2	HEM 00		14	30F 031	
3	CEG 031		15	PRS 0040	
4	CEG 032		16	PRS 0037	
5	OOT 01		17	PRK 0020	
6	HEM 01		18	-11- 03114	
7	HEM 01		19	-() - 0000	
8	-1/1- 030 16		20	-11- 03116	
9	PRM 030		21	HEM 02	
10	PRK 0010		22	HEM 02	
11	KET 01		23	CTR 036	
12	HEM 00		24	30C 037	
			25	CEG 051	
			26	HEM 00	
			27	KET 77	
			28	KPM 77	

Продолжение

Номер слова	Содержание в восьмеричном формате
143	000 003
144	000 004
145	000 005
000	377 377

Программа

Номер инструкции	Мнемоника	Примечание	Номер инструкции	Мнемоника	Примечание
0000	НСТ 00		0008	ПРК 0010	
0001	НБИ 00		0009	НСТ 01	
0002	СБС 031		000a	НБИ 00	
0003	ОСТ 01		0011	-11- 032 16	
0004	ПБИ 01		0012	ЧФ 031	
0005	НБИ 01		0013	ПРС 0140	
0006	-1/1- 032 16		0014	ПРС 0000	
0007	ТВД 032		0015	ПРК 0006	
			0016	-11- 03114	
			0017	-()- 00214	
			0018	НСТ 77	
			0019	НПН 77	

(бит вкл. при наличии опко-ки)

После включения программы:

Время с начала выполнения программы	Информация в слове 000
1 с	000 000
2 с	000 001
3 с	000 002
4 с	000 003
5 с	000 004
6 с	000 005

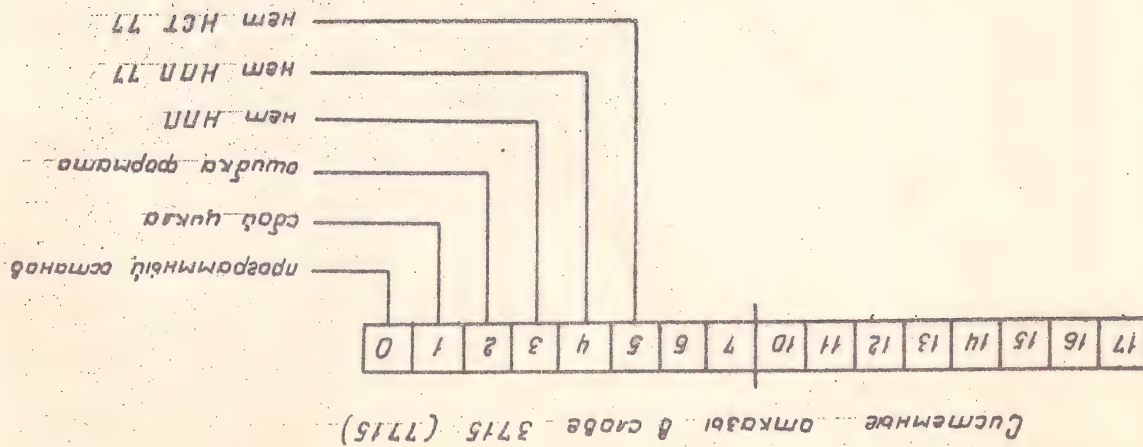


Рис. 2.15

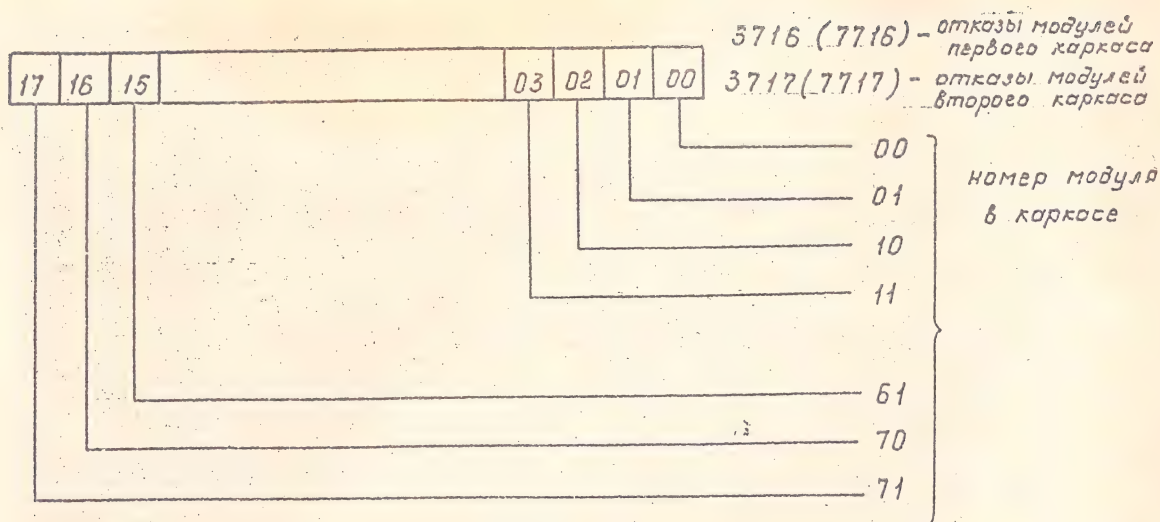


Рис. 2.16

Номер инструкции	Мнемоника	Примечание
0008	ПРК 0020	
9	НСТ 01	
10	НВН 00	
11	-11- 0116	- через каждые 2с бит принимает истинное значение
12	ПФФ 010	
13	ПРС 0030	начало массива А
14	ПРС 0100	начало массива В
15	ПРК 0008	количество пересчитанных слов
16	ПРК 0001	одно слово за один скан
17	-11- 01014	
18	ЗСТ 77	
19	-11- 010 16	
20	-() - 000 02	- бит возбуждается через 16с
21	НСТ 77	
22	НПН 77	
23		

Переименование информации первого файла во второй происходит по одному слову через каждые 2 с.

ЧИТАТЬ СЛОВО ИЗ ФАЙЛА

ЧСФ ХХХ - адрес счетчика

ПРС ХХХ - начальный адрес файла

ПРС ХХХ - адрес слова

ПРК ХХХ - длина файла

При изменении состояния запускаемого выражения из "0" в "1"

происходит переименование содержимого слова файла по адресу слова.

При следующем изменении запускаемого выражения из "0" в "1"

содержимое следующего слова файла переименовается по тому же адресу слова.

Пример.

Исходная информация:

Номер слова	Содержимое в восьмеричной форме
140	000 000
141	000 001
142	000 002

где L - длина файла, равная количеству слов файла;
 V - количество слов за скан, равное скорости переи-
 щения (Рис. 2.23).

Выполнение инструкции осуществляется пересылкой количества слов за один скан, равное скорости переищения. Перемещение про-
 изводится слово в слово. Содержимое счетчика будет увеличиваться
 после каждого переищения, а перемещение происходит при измене-
 нии заускающего выражения из 0 в 1. Как только содержимое счет-
 чика станет равным длине файла L , бит 16, характеризующий вы-
 полнение инструкции, возобудится.

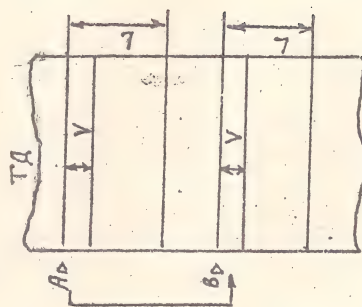


Рис. 2.23

Пример.
 Программа

Номер инструкции	Имякода	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	ОСТ 01	
3	СВС 010	
4	ПБЛ 01	
5	НБЛ 01	
6	-1/1-011 16	
7	ТВД 011	

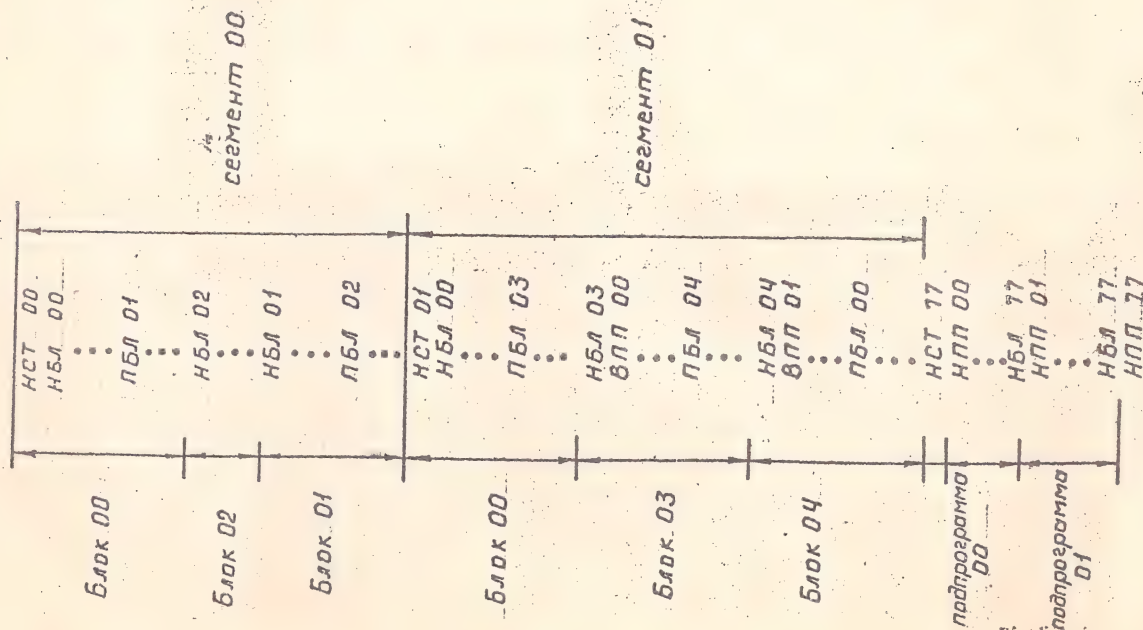


Рис. 2.17

Программируемый контроллер
Программа
Лист

Номер инструкции	Мнемона	Примечание	Номер инструкции	Мнемона	Примечание
0			0		
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		
0			0		
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		
0			0		
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		
0			0		
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		

Рис. 2.18

Продолжение

Номер инструкции	Мнемона	Примечание
00 17	-(-)-000 03	
18	-Ш- 000 03	
19	-(-)-000 02	
20	НСТ 77	
21	НПН 77	

В результате выполнения программы, через 2с включится бит 000 02.

2.11. Файловые инструкции

Файловые инструкции - это такие инструкции, которые оперируют с группами слов ТД.

Файл определяется как группа последовательных слов ТД, используемых для хранения информации. Каждая файловая инструкция использует счетчик. Слово, на котором работает счетчик файловой инструкции имеет формат:

17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0
----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---

накопленное значение

бит ошибки в адресации

бит завершения файловой инструкции

бит возбуждения

Если запускаемое выражение равно 0 и введен 16 бит, то счетчик обнуляется. Все файловые инструкции условные. Состояния файловых инструкций: возбуждение, завершение и ошибка отображаются соответственно следующими битами 17, 16, 14 слова счетчика, связанного с данной файловой инструкцией.

ПЕРЕМЕННЫЕ ФАЙЛ

ПФ 1111, где 1111 - адрес счетчика.

Перемещаемый файл задается параметрами:

ПФХХХХХХ - адрес счетчика;

ПРСХХХХХХ - начальный адрес файла А;

ПРСХХХХХХ - начальный адрес файла В;

ПРХХХХХХ - длина файла L;

ПРХХХХХХ - скорость перемещения V;

Пример.

Необходимо сравнить содержимое по адресу 010 с содержимым по адресу 011, если равны, то включить 000 00 бит, если не равны, то сравнить на "меньше". При истинности инструкции, включить бит 000 01, при ложном состоянии выполнить поддержку времени 2 с, прочесть содержимое по адресу 010 и сравнить на принадлежность пределам 4095 и 1000. При истинном результате, включить бит 000 03. При ложном результате - включить бит 000 02.

Исходные данные:

Номер слова	Содержимое в десятичном формате
010	2048
011	4095
012	1000

Программа:

Номер инструкции	Мнемикод	Примечание
0000	HST 00	
1	HBL 00	
2	RBN 010	
3	PRS 0011	
4	-() - 000 00	
5	MNH 010	
6	PRS 0012	
7	-() - 000 01	если равно и больше бит не включается
8	-И- 000 01	
9	TBC 003	
10	PRK 0002	
11	-11- 003 16	
12	HBL 01	
13	HBL 01	
14	PRH 010	
15	PRS 0011	
16	-PRC 0012	

торых отмечается инструкцией "начало сегмента".

В свою очередь каждый сегмент программы состоит из блоков, начало каждого из которых отмечается инструкцией "начало блока". Максимально возможное число различных сегментов программы равно 8 (от 0 до 7) и максимально возможное число различных блоков в одном сегменте равно - 768. Максимально возможное число подпрограмм равно восьми.

Количество сегментов в программе, количество блоков в сегменте, а также порядок следования блоков внутри сегмента могут быть произвольными. В частном случае, весь сегмент может быть одним блоком, а вся программа - одним сегментом. Программа может состоять из одного сегмента и одного блока.

Каждый сегмент (за исключением нулевого сегмента) может начинаться в одном из трех состояний: вычисление, игнорирование, обнуление.

Вычисление - значит программа активного блока выполняется.

Игнорирование - программа сегмента не выполняется.

Обнуление - процессор проходит инструкцию сегмента с результатом равным нулю в течение одного скана.

Нулевой сегмент всегда вычисляется (прибыль питания этого сегмента обнуляется).

При первом проходе программы все сегменты находятся в обнулении - это значит, что процессор проходит программу с результатом равным нулю и поэтому все выходы, включенные в программу, обнуляются, а слова, на которых запущены таймеры на включение - сбрасываются. После команды HST 77, все сегменты устанавливаются в режим игнорирования.

В следующих проходах процессор при переходе от сегмента к сегменту пользуется информацией области меток активных блоков (область задания текущего состояния на рис. 2.13). В эту область заносится информация, после каждой команды структурирования.

Информация о текущем состоянии сегмента занимает 1 байт.

В шесть младших разрядов (0-5) записывается номер активного блока, который начнет работать в следующем скане.

В шестой и седьмой разряд записываются биты информации, после выполнения инструкции структурирования: OST, 3ST, 5ST.

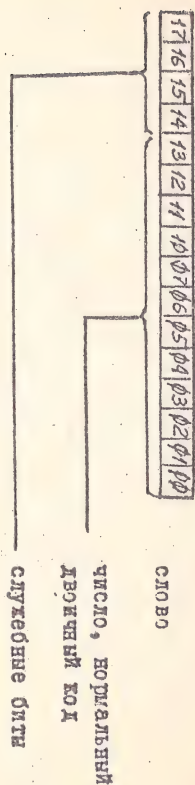
Каждый блок сегмента может находиться в одном из двух состояний: активном или пассивном. На протяжении одного цикла сканирования программы в каждом сегменте может быть только один активный блок. При включении питания ПК во всех сегментах программы устанавливаются активными нулевые блоки.

2.4. Представление числа

Содержимое слова (т.е. комбинация состояний его бит) может подвергаться арифметической обработке. В этом случае оно имеет следующий формат:

биты 00, 01 - 13 - число в нормальном двоичном коде;
биты 14-17 - служебные.

Состояние служебных битов приобретает различное значение в зависимости от выполняемой инструкции:



Таким образом, в контроллере допускается непосредственная обработка чисел в интервале от 0 до 4095. 4095 - максимальное значение установки таймера. ПК обрабатывает только положительные числа.

Содержимое одного слова можно также закодировать восьмеричными числом:

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	номер бита
2 ¹	2 ⁰	2 ³	2 ²	2 ¹	2 ⁰	2 ³	2 ²	2 ¹	2 ⁰	2 ³	2 ²	2 ¹	2 ⁰	2 ³	2 ²	2 ¹	2 ⁰	всего 8 бит
1	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	1	восьмеричное число
2	3	1	3	5	7													

Биты 00-07 содержат восьмеричное число младшего байта. Биты 08-17 содержат восьмеричное число старшего байта.

2.5. Инструкции логические

Эта группа инструкций используется для опроса состояния битов ТД - входов или выходов, служебных битов таймеров, счетчиков, а также для управления состоянием битов выходов.

Логические инструкции подразделяются на адресные и безадресные.

2.10. Инструкции сравнения

Эта группа инструкций позволяет производить сравнения в рабочей программе. Инструкции работают с данными в формате целых чисел и в пределах от 0 до 4095.

Инструкции сравнения являются входными инструкциями. Если цепь содержит только инструкции сравнения и выходная инструкция выключается, при ложном состоянии инструкции сравнения, несохраняемая выходная инструкция выключается. Если цепь содержит другие входные инструкции, кроме сравнения, то выходная инструкция выключается, если можно отследить хотя бы один замкнутый путь из логической истинных инструкций. Если же такого пути нет, то несохраняемая выходная инструкция выключается, а сохраняется - остается в прежнем состоянии.

СРАВНИТЬ НА РАВНО

РВН *л л л*, РВН *л л л*

ПРС *л л л л л*, ПРК *к к к к*

Если содержимое по адресу слова *л л л* равно содержимому по адресу *л л л л л*, то инструкция логически истинна, если же данные не равны, то инструкция ложна. Если ПРК-то сравнение с константой.

СРАВНИТЬ НА ПРИНАДЛЕЖНОСТЬ ПРЕДЕЛАМ

ПРН *л л л*, ПРН *л л л*

ПРС *л л л л л*, ПРК *л л л л л*

ПРС *к к к к к*, ПРК *к к к к*

Если число, хранящееся по адресу *л л л* одновременно больше или равно числу, записанному по адресу *к к к к* и меньше или равно числу, записанному по адресу *л л л л л*, то инструкция ПРН истинна. Если после инструкции-ПРК, то сравнение в том же порядке, только с константами.

СРАВНИТЬ НА МЕНЬШЕ

ПРН *л л л*, МНШ *л л л*

ПРС *л л л л л*, ПРК *к к к к*

Если число, хранящееся по адресу *л л л* меньше, чем число записанное по адресу *л л л л л*, то инструкция ПРН истинна, в противном случае - логически ложна.

СРАВНИТЬ НА БОЛЬШЕ

БШШ *л л л*, БШШ *л л л*

ПРС *л л л л л*, ПРК *к к к к*

Если число, хранящееся по адресу *л л л* больше, чем число, записанное по адресу *л л л л л*, то инструкция БШШ логически истинна, в противном случае - ложна. Инструкции сравнения с ПРК работают только в МС 59.07.

Адрес бита должен находиться в пределах 000 00 - 377 178.
 Структура адреса бита - см. рис. 2.4.
 Операторы адресных инструкций на языке релейно-контактных

символов:

- II- - прямой опрос входа;
- И- - инверсный опрос входа;
- ()- - включить выход;
- (S)- - включить выход и зафиксировать;
- (R)- - выключить выход и зафиксировать;

Операторы безадресных инструкций:

- I- - начало одного ветвления;
- И- - конец одного ветвления;
- И- - перенести маркер в начало ветвления.

Операторы адресных инструкций на языке булевой алгебры:

- ' - конъюнкция;
- / - конъюнкция с инверсией;
- - присвоить;
- = S - присвоить "1" и зафиксировать;
- = R - присвоить "0" и зафиксировать.

Операторы безадресных инструкций:

- (- скобка открывающая ;
-) - скобка закрывающая ;
- + - дизъюнкция.

ПРЯМОЙ ОПРОС ВХОДА

Когда входное устройство находится в замкнутом состоянии, как например, включенный конечный выключатель,

индицируется включенное состояние. Это состояние передается в соответствующий бит памяти.

Когда в программе встречается инструкция опроса на включение с этим адресом, процессор определяет, что входное устройство включено и считает инструкцию истинной. Если входное устройство замыкается, процессор считает эту инструкцию ложной.

Если цепь содержит только эту инструкцию и выходящую инструкцию, то входная инструкция возбуждается (бит включается), когда входная инструкция истинна.

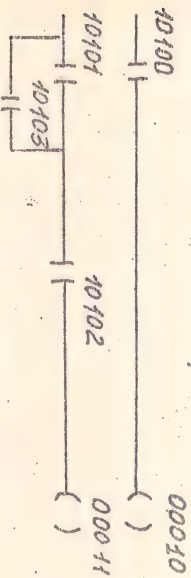
Если же входная инструкция ложна (вход разомкнут), то нейтральный выход выключается.

Если в цепи содержится еще входные инструкции, то выходная инструкция возбуждается, когда может быть найден по крайней мере

Номер инструкции	Мнемоника	Примечание
0005	НБЛ 01	
0006	-I/I- 003 I6	
0007	ТВД 003	
0008	ПРК 0010	
0009	-II- 003 I6	
0010	СДМ 000	
0011	-II- 000 00	
0012	ПВЛ 02	
0013	НБЛ 02	
0014	ЧТС 012	
0015	ЗПС 000	
0016	ПБЛ 01	
0017	НСТ 77	
0018	НШ 77	

I шаг	I7	I6	I5	I4	I3	I2	I1	I0	7	6	5	4	3	2	I	0	- (000)
2 шаг		I															
3 шаг			I														
I6 шаг																	I
I7 шаг		I	I	I	I	I	I	I	I								
I8 шаг		0	I	I	I	I	I	I	I	I							
		0	0	0	0	0	0	0	0	0	I	I	I	I	I	I	I

Один замкнутый путь из логически истинных инструкций, ведущих от шины питания к выходной инструкции. Если такого пути не существует, то незафиксированные выходы отключаются. Пример двух цепей:

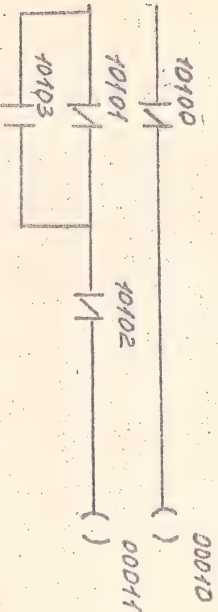


Выход 000 10 возбуждается, если контакт 101 00 замкнут. Выход 000 11 возбуждается, если контакты 101 01 и 101 02 замкнуты, либо замкнуты контакты 101 03 и 101 02.

ИНВЕРСНЫЙ ОПРОС ВХОДА

Если входное устройство не замкнуто, то выходной зажим, к которому это устройство присоединено, инициирует выключенное состояние. Когда в программе встречается инструкция инверсного опроса, процессор определяет, что входное устройство выключено и считает инструкцию истинной. Если входное устройство замкнуется, то процессор считает инструкцию ложной.

Если цепь содержит только инструкцию инверсного опроса и выходную инструкцию, то выходная инструкция включается, когда инструкция инверсного опроса находится в истинном состоянии, т.е. вход отключен. Если инструкция ложна, то несохраняемый выход отключается. Пример двух цепей:



Выход 000 10 включается, если контакт 101 00 разомкнут.

Выход 000 11 включается, если контакты 101 01 и 101 02 разомкнуты, либо 101 03 замкнут, а 101 02 разомкнут.

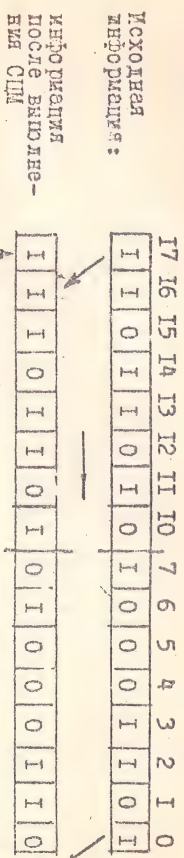
Бит информации переходит на место младшего бита. Затем, все оставшиеся биты в содержимом сдвигаются влево или вправо на один разряд.

СДВИГ ЦИКЛИЧЕСКИЙ К МЛАДШИМ БИТАМ

СМ $n \ n \ n$

При выполнении инструкции СМ содержимое по адресу слова сдвигается на один бит в сторону младшего разряда.

Пример.

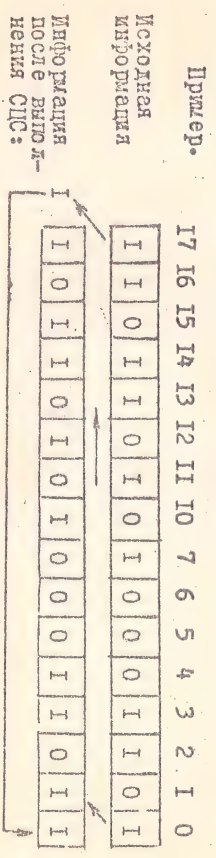


СДВИГ ЦИКЛИЧЕСКИЙ К СТАРШЕМУ БИТУ

СМ $n \ n \ n$

При выполнении инструкции СМ содержимое по адресу слова сдвигается на один бит в сторону старшего разряда.

Пример.



Информация после выполнения СМ:

Пример.

Через 10 включать поочередно биты в слове 010.

Исходная информация: в слове (011) записать константу 200 000₈ и в слово (012) - 377 000₈

Программа

Номер инструкции	Мнемоника	Примечание
0000	ИСТ 00	
0001	ИВЛ 00	
0002	ЧИС 011	
0003	ЗНС 000	
0004	ИВЛ 01	

Исходная информация

Номер слова	Содержимое в восьмеричном формате
010	111 111
011	222 222
012	333 333
013	000 000

Программа

КСТ 00
 НВ1 00
 ЧВ1 010
 ЗВ1 012
 ЧВ2 011
 ЗВ2 012
 ЧВ3 012
 ЗВ3 013
 НВ2 013
 КСТ 77
 НВ1 77

Информация, после выполнения программы:

Номер слова	Содержимое в восьмеричном формате
010	111 111
011	222 222
012	222 111
013	111 222

Инструкция сдвига бита

При помощи этих инструкций можно организовать битовый сдвиг. При этом сдвиг внутри восьмеричного слова.

Инструкция сдвига бита является выходной. Когда цепь перешла из ложного в истинное состояние, инструкция начинает работать.

Инструкция действует следующим образом.

Сначала один бит выводится из сдвигаемого содержимого по адресу слова. Если сдвиг в сторону младшего разряда, то бит, переходящий на место старшего. При сдвиге в сторону старшего разряда, старший

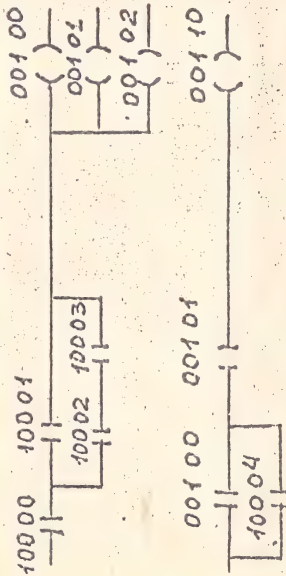
ВЫХОДНЫЕ ИНСТРУКЦИИ

Инструкция выведения выхода используется для включения определенных битов памяти. Если адрес бита соответствует адресу записи выходного модуля, то выходные устройство, соединенное с этим выходом, также включается. Состояние включения или выключения выходного бита определяется логическим состоянием цепи в рабочей программе пользователя.

Если в цепи существует замкнутый путь из логически истинных выходных инструкций, то выходная инструкция возбуждается (включается). Если такого пути нет, то условия цепи ложны и выходная инструкция выключается. Если условия цепи ложны, то выходное устройство отключается.

Инструкция выведения выхода подобна катушке реле. Выходная инструкция управляет предшествующим ей входным инструкциям в программной цепи. Катушка реле управляет контактами в электрической цепи. Замкнутый путь из логически истинных инструкций соответствует электрической цепи из замкнутых электрических контактов.

Любой бит ТД в пределах 0-3778 слов может быть опровергнут любой инструкцией опроса столько раз, сколько необходимо. Таким образом, бит управляемый выходной инструкцией, также может быть опровергнут необходимым количеством раз, например:



Биты 00100 и 00101 в первой цепи - выходные. Эти же биты во второй цепи участвуют во входных инструкциях.

ФИКСАЦИЯ И РАСКЛЮЧЕНИЕ ВЫХОДА

Инструкция фиксации и расключения выхода является сохраняющей выходную инструкцией. Они обычно используются в паре для управления битом ТД. Если адрес инструкции фиксации соответствует адресу записи выходного модуля, то выходное устройство, со-

дленное с этим значением включается при включении бита, что определяется условием его цепи.

Если в цепи может быть найден замкнутый путь из логически истинных инструкций, то инструкции фиксации выхода включает управляемый ед бит. Когда условия в цепи перестают быть истинными, бит, управляемый инструкцией фиксации остается включенным, если он перед этим был включен.

Инструкция расфиксации, включая тот же адрес, что и инструкция фиксации, используется для выключения бита памяти. Когда в цепи устанавливается замкнутый путь из логически истинных инструкций, инструкция расфиксации выключает управляемый ед бит.

Когда ПК переходит из режима РАБОТА в режим ПРОГРАММИРОВАНИЯ, или происходит потеря питания, порождаемая из включенных инструкций фиксации или расфиксации сохраняется состояние управляемого ед бита. При переходе в режим РАБОТА выходное устройство, соответствующее зафиксированному биту, сразу включается, даже если условия в цепи ложны.

Программа пользователя может опрашивать бит, управляемый инструкцией фиксации и расфиксации, произвольное количество раз.

Например, включение катушки с фиксацией:



При замыкании контакта 11000 в память по адресу 000 00 заносится логическая 1, зачитается светодиодом 01314 и 01315. При размыкании контакта 110 00 светодиод 01314 гаснет, а светодиод 01315 продолжает гореть. Если произойдет потеря питания, а затем его восстановление, светодиод 01315 не гаснет, то есть логическая единица по адресу 000 00 сохраняется в памяти.

Программа для реализации цепи с фиксированной катушкой: Перед вводом программы необходимо задать: 30, ПРП, время цикла и указать места расположения модулей. Номер инструкции пользователя вводить не нужно. Вводится только mnemonic.

ЧИС 012
ПСС 013
ЗПС 014
НСТ 77
ННП 77

Номер слова	Содержимое в восьмеричном формате
012	324 017
013	123 226
014	170 146
	- результат

ЧИТАТЬ БАЙТ МЛАДШИЙ

ЧМ 777

При выполнении инструкции ЧМ младший байт содержимого по адресу слова пересылается в младший байт аккумулятора.

ЧИТАТЬ БАЙТ СТАРШИЙ

ЧС 777

При выполнении инструкции ЧС старший байт содержимого по адресу слова пересылается в старший байт аккумулятора.

ЗАПИСАТЬ БАЙТ МЛАДШИЙ

ЗМ 777

При выполнении этой инструкции содержимое младшего байта аккумулятора пересылается в младший байт указанного адреса.

ЗАПИСАТЬ БАЙТ СТАРШИЙ

ЗС 777

При выполнении этой инструкции содержимое старшего байта аккумулятора пересылается в старший байт указанного адреса.

ОБМЕНЯТЬ СОДЕРЖИМЫ БАЙТ СТАРШИЙ И БАЙТ МЛАДШИЙ

ПБ 777

При выполнении этой инструкции содержимое байта старшего пересылается в байт младший, а содержимое младшего байта в старший.

Пример.

Необходимо младший байт слова (010) переслать в младший байт слова (012), а старший байт содержимого слова (011) переслать в старший байт (012), в (013) слово записать содержимое (012) слова, только байт старший пересылается в младший, а младший байт в старший.

Информация, после выполнения программы:

Номер слова	Содержимое в восьмеричном формате
OIO	I23 000
OII	I23 000

ПОБИТНОЕ УМНОЖЕНИЕ СЛОВ

ПУС ппп

Если условия в цепи истинны, то над каждой парой одноименных битов из аккумулятора и содержимого по адресу слова, производится логическая операция И.

Результат записывается в аккумулятор.

Пример.

ИСТ 00
НБЛ 00
-II- 100 00
ЧИС 012
ПУС 013
ЗПС 014
ИСТ 77
НПШ 77

Номер слова	Содержимое в восьмеричном формате	Исходная информация
012	324 017	- Результат выполнения инструкции ПУС
013	135 302	
014	124 002	

ПОБИТНОЕ СРАВНЕНИЕ СЛОВ

ПСС ппп

Если условия в цепи с инструкцией ПСС истинны, то над каждой парой бит из аккумулятора и содержимого по адресу слова производится логическая операция побитного сравнения, результат записывается в бит аккумулятора с тем же номером.

Пример.

ИСТ 00
НБЛ 00
-II- 100 00

Номер инструкции	Имякокод	Примечание
0000	ИСТ 00	
1	НБЛ 00	
2	-II- 110 00	
3	-(S)-00000	
4	-()- 01314	
5	-II-00000	
6	-()-01315	
7	ИСТ 77	
8	НПШ 77	

РАСЧИСЛЕНИЯ ВЫХОДА

При замыкании контакта I1000 (рис. 2.19) загорается светодiod 01314, а светодiod 01315 не горит.

При размыкании контакта I10 00 светодiod 01314 гаснет.

При выключении питания, а затем при его восстановлении светодiodы 01314, 01315 не горят.

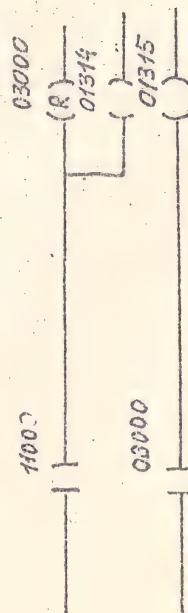


Рис. 2.19

Программа для цепей, изображенных на рис. 2.19:

Номер инструкции	Имякокод	Примечание
0000	ИСТ 00	
1	НБЛ 00	
2	-II- 11000	
3	-(R)- 03000	
4	-()- 01314	
5	-II- 03000	
6	-()-01315	
7	ИСТ 77	
8	НПШ 77	

ИНСТРУКЦИИ ВЕТВЕВНИК

Группа инструкций ветвления выделяет три инструкции:

- начало одного ветвления;
- конец одного ветвления;
- переход маркера в начало ветвления.

Эти инструкции служат для программирования параллельных логических ветвей в ПРП.

Часто входное устройство должно включаться при более, чем одном сочетании входных условий. Инструкции ветвления применяются для формирования параллельных ветвей, формирующих более, чем одно условие включения выходной инструкции. Для включения выходной инструкции достаточно, чтобы логически замкнутый путь был установлен хотя бы в одной из параллельных ветвей.

Другими словами, между условиями параллельных ветвей производится логическая операция ИЛИ.

Если же ни в одной из ветвей не установлен замкнутый путь, то условия в цепи коны и выход не включается.

На рис. 2.20 представлена цепь, являющаяся разветвленное условием и входной инструкцией.

В этой цепи вход включается при включении любого из входных битов: 101 00 или 101 01. Как показано на рисунке логическое замкание цепи может произойти по любому из двух путей, по любому из двух параллельных ветвей.

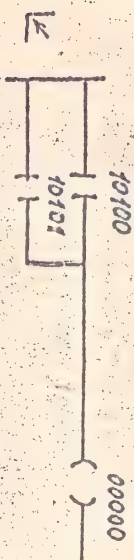


Рис. 2.20

На рисунке 2.20 изображена цепь реализующая логическую операцию ИЛИ: 101 00 V 101 01 = 000 00.

Используя инструкции ветвления, можно программировать одну или несколько параллельных ветвей:

СЧЕТЧИК БИТОВ

СЧП ППП СЧП ППП
ПРК КККК ПРК ММММ

Инструкция СЧП увеличивает в каждом слове программы содержимое слова, ППП, если результат равен 1, и оставляет содержимое этого слова без изменения, если результат равен 0.

Инструкция СРС сбрасывает содержимое счетчика в ноль.

Если значение счетчика достигло значения условия, включается бит 16.

2.9. Инструкции для работы с данными

Все инструкции основной обработки являются условными и входными. Если условия в цепи истинны, то инструкции выполняются. Эти инструкции используются для переноса данных из одного слова таблицы данных в другое слово, для копирования накопленных значений таймеров и счетчиков, для включения отдельных битов.

ЧИТАТЬ СЛОВО

ЧПС ПП

адрес слова

При выполнении инструкции ЧПС содержимое по адресу слова записывается в аккумулятор.

ЗАПИСАТЬ СЛОВО

ЗПС ПП

При выполнении инструкции ЗПС содержимое аккумулятора записывается по адресу слова.

Пример.

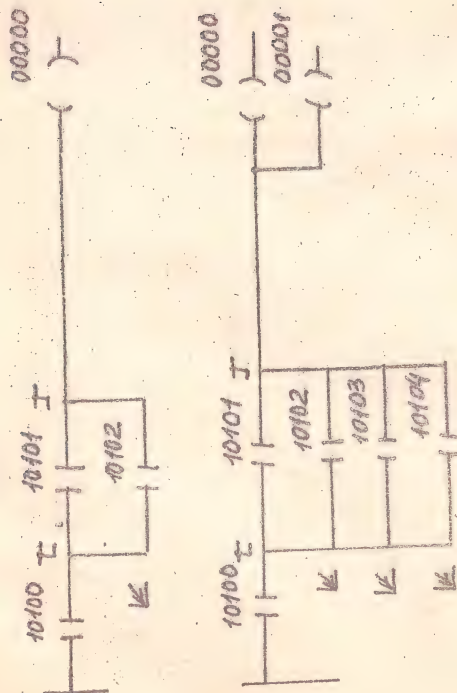
Исходная информация:

Номер слова	Содержимое в восьмеричной форме
010	123 000
011	000 000

Программа

ПСТ 00
НМН 00
-11- 100 00
ЧПС 010
ЗПС 011
ПСТ 77
НМН 77

Номер инструкции	Мнемоник	Примечание
0003	СВС 004	
4	ПВЛ 01	
5	НВЛ 01	
6	-1/1-001 I6	
7	ТВЛ 001	
8	ПРК 0010	
9	-11-001 I6	
10	ПВЛ 02	
11	НВЛ 02	
12	-1/1-003 I6	
13	ТВЛ 003	
14	ПРК 0020	
15	-11-003 I6	
16	ПВЛ 01	
17	НСТ 01	
18	НВЛ 00	
19	-11-001 I6	
20	СЧС 004	
21	ПРК 000 8	
22	-11-004 I6	
23	-(R)-004 I6	
24	ПВЛ 01	
25	НВЛ 01	
26	-11-003 I6	
27	С4В 004	
28	ПРК 0003	
29	-11-004 I6	
30	-(R)-004 I6	
31	ПВЛ 00	
32	НСТ 77	
33	НПШ 77	



Можно программировать выключенные параллельные ветви, т.е. ветви, параллельные части несоединенной ветви (рис. 2.21).

В ПК допускается обработка цепей, имеющих до восьми вложенных. Если программа в ПК вводится с помощью МВ 96.02, то цепи должны содержать не более пяти вложенных и последовательно включенных контактов до 8.

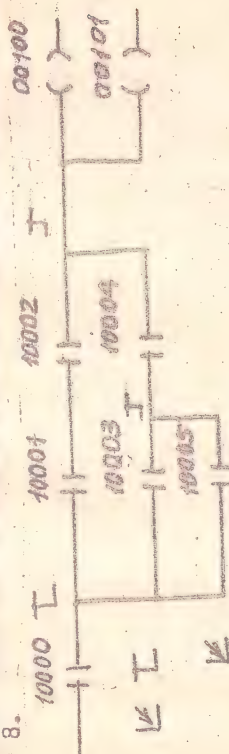


Рис. 2.21

Программа, реализующая цепь, изображенную на рис. 2.21:

Счетчик на сложение используется для счета переходов цепи из ложного в истинное состояние. Эти переходы могут быть вызваны прохождением деталей перед детектором или включением конечного выключателя. Отсчет сохраняется при переходе цепи в ложное состояние, сброс счетчика происходит после выполнения инструкции СЕС, итершей тот же адрес, что и счетчик.

Когда цепь с инструкцией счета переходит из ложного в истинное состояние, накопленное значение увеличивается на единицу. Когда накопленное значение достигнет уставки, возбуждается 16 бит слова, на котором запущен счетчик на сложение.

Бит 15 служит битом разрешения счета на сложение. Он включается, когда истинны условия цепи.

Инструкция счета на сложение предназначена для счета до своей уставки. Если счет продолжается, после достижения уставки, а накопленное значение переходит в свое предельное значение 4095, то включается бит 14. Биты 14 и 16 сбрасываются инструкцией СЕС с тем же адресом, что и инструкция счетчика на сложение.

Пример программы с использованием счетчика на сложение.

Исходные данные: слово 0020 записать константу 0010₁₀.

Номер инструкции	Имя кода	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	ОСТ 01	
3	СЕС 000	
4	НБЛ 01	
5	НБЛ 01	
6	-1/1- 010 16	
7	ТВД 010	
8	ПРК 0001	
9	-11- 010 16	
10	-() - 012 00	
11	НСТ 01	
12	НБЛ 00	
13	-11- 010 16	
14	СЧ 000	
15	ПРС 0020	
16	НСТ 77	
17	НПШ 77	

Инструкция определяет начало блока в данном сегменте. Все следующие за ней инструкции (за исключением инструкции НСТ, НБЛ выполняются при условии, если данный блок активен). Информации о том, какой блок активен в данном сегменте процессор получает из области текущего состояния. Инструкция НБЛ устанавливает результат в единицу. Это значит, если за инструкцией НБЛ следует инструкция, включить выход, то данный выход возбуждается.

Последовательность блоков в сегменте может быть произвольной.

ПЕРЕЙТИ НА БЛОК

НБЛ номер блока

Когда цепь, содержащая инструкцию НБЛ, становится истинной, процессор записывает в область текущего состояния, в байт соответствующий обрабатываемому сегменту, номер активного блока, указанный в операндной части инструкции НБЛ. Следующие за НБЛ инструкции не выполняются.

Когда цепь, предшествующая НБЛ, ложна, процессор не выполняет инструкцию НБЛ, а выполняет следующие за ней инструкции.

На рис. 2-22 приведен пример, поясняющий принцип сканирования.

ЗАКРЫТЬ СЕГМЕНТ

ЭСТ номер сегмента

Когда цепь, предшествующая ЭСТ, истинна, процессор устанавливает в состоянии игнорирования тот сегмент, который указан в операндной части инструкции (за исключением нулевого сегмента).

Если ЭСТ имеет в операндной части номер сегмента тот же, что и номер сегмента в котором находится данная инструкция, то на протяжении одного скана, в котором инструкция ЭСТ выполнялась, инструкции следующие за ней так же будут исполняться. В следующем скане, процессор не выполнит ни одной инструкции данного сегмента и так будет продолжаться до тех пор, пока не выполнится инструкция ЭСТ с тем же номером. Вычисления в сегменте начнутся в блоке, который работал в момент выполнения инструкции ЭСТ.

ОТКРЫТЬ СЕГМЕНТ

ОСТ номер сегмента

Когда цепь, содержащая инструкцию ОСТ, становится истинной, то сегмент, указанный в операндной части, устанавливается в состояние вычисления. В байте области текущего состояния данного

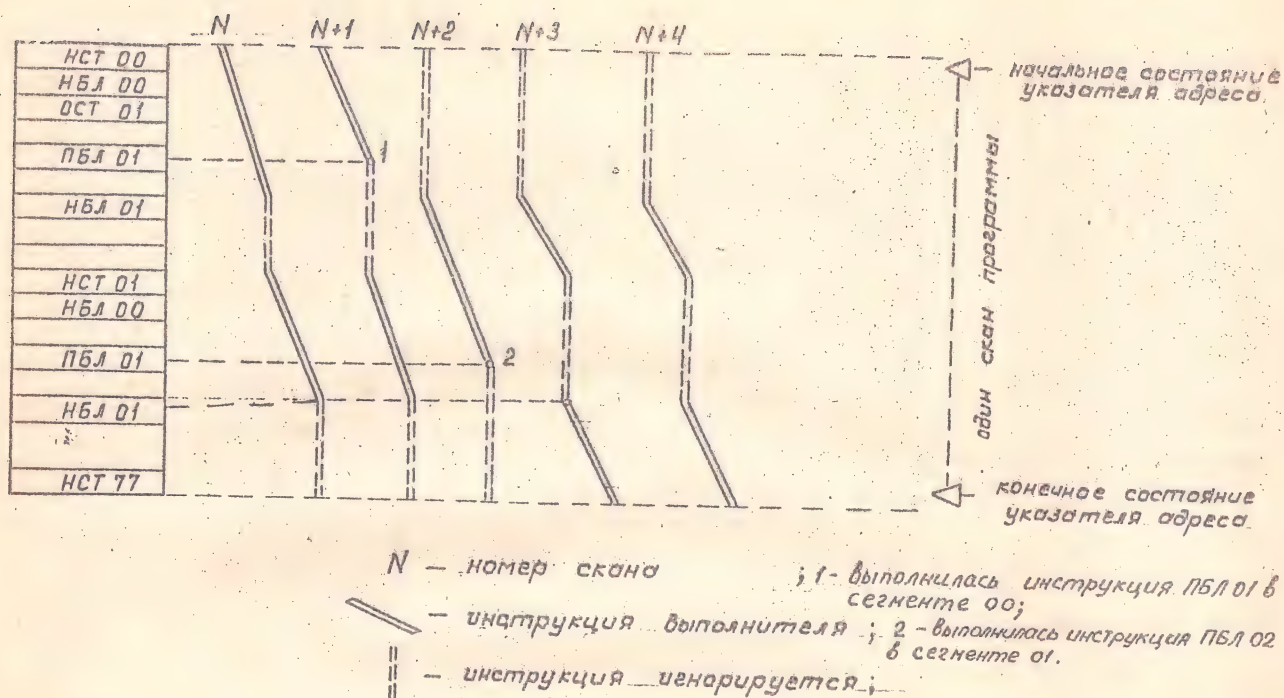


Рис. 2.22

Номер инструкции	Мнемикод	Примечание
0008	ПРК 0010	
0009	-11-001 16	
0010	ПВЛ 02	
0011	НВЛ 02	
0012	-11-003 16	
0013	ТВЛ 003	
0014	ПРК 0020	
0015	-11-003 16	
0016	ПВЛ 01	
0018	НСТ 01	
0019	НВЛ 00	
0020	-11-003 16	
0021	ТВЛ 000	
0022	ПРК 0300	
0023	-11-000 16	
0024	ПВЛ 01	
0025	НВЛ 01	
0026	-11-002 16	
0027	ТВЛ 002	
0028	ПРК 0040	
0029	-11-002 16	
0030	ПВЛ 00	
0031	НСТ 77	
	НПН 77	

СЧЕТЧИК НА СЛОЖЕНИЕ

СЧС ппп

ПРС ттт тт,

СЧС ппп

ПРК ккк к.

Если в период работы накапливающего таймера условия в цепи становятся ложными, то это не приводит к потере накопленного значения. Например, когда цепь снова станет истинной, таймер продолжит счет времени с той величины, которая была накоплена к моменту останова. Благодаря способности сохранять накопленное значение этот таймер может измерять суммарное время истинного состояния его цепи.

Накопленное значение накапливающего таймера может быть сброшено инструкцией сброса (СБС) с тем же адресом, что и таймер.

Бит 16 включается, когда накопленное значение достигает установленного. Этот бит может быть сброшен только инструкцией сброса.

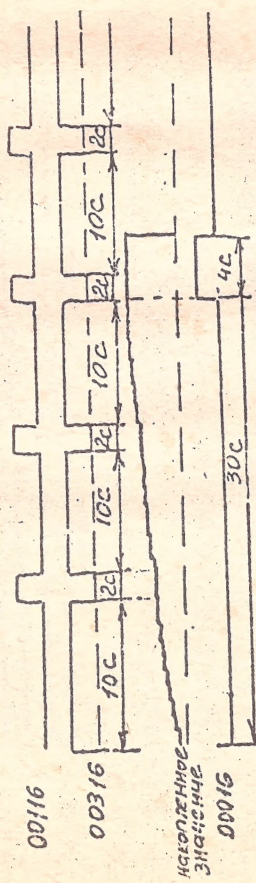
Бит 17 включается, когда условия в цепи становятся истинными и включается, когда эти условия становятся ложными.

Бит 15 отображает дискрету времени.

Пример.

Через 20 с суммарного времени истинного состояния цепи, выдать сигнал длительностью 4с.

Временная диаграмма



Программа

Номер инструкции	Мнемоника	Примечание
0000	ИСТ 00	
0001	НЕМ 00	
0002	ОСТ 01	
0003	СБС 000	
0004	НЕМ 01	
0005	НЕМ 01	
0006	-1/1-001 16	
0007	ТЭС 001	

сегмента бит 11 шесть и бит семь принимают истинное значение.

СБРОСИТЬ СЕГМЕНТ

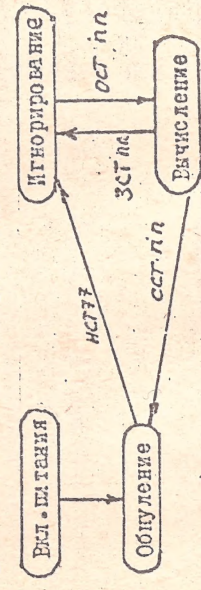
ИСТ nn номер сегмента

Когда цепь, содержащая инструкцию ССТ, становится истинной, то сегмент, указанный в операндной части, устанавливается в состояние обнуления, т.е. процессор исполняет инструкции независимо от вычислений, с результатом равным нулю. Бит текущего состояния становится равным нулю.

После выполнения команды ИСТ 77, сегмент устанавливается в состояние игнорирования. Инструкции ССТ необходимо писать в программе до начала соответствующего сегмента, иначе обнуления не произойдет.

Из состояния игнорирования, сегмент выводится командой ССТ и вычисление начинается с нулевого блока.

Взаимосвязь команд структурирования:



nn -- номер сегмента

УСТАНОВИТЬ НОМЕР СЕКТОРА

УНС nn номер сектора

Отказы модулей фиксируются в словах 3716(7716 - для MS9.06) и 3717(7717 - для MS9.07). Чтобы получить доступ в слово, например, 3716, необходимо записать:

```

:
УНС 07
ЧТС 316
УНС 00
ЗПС W
:
W -- слово, в котором
будут анализироваться
биты отказов.

```

Для доступа в слово 7716, необходимо записать:

УНС 17
 ЧТС 316
 УНС 00
 ЗНС W

2.7. Инструкции для работы с подпрограммами

Однократные фрагменты в программе можно организовать, как одну подпрограмму и вызывать в нужные моменты. Обращения к подпрограммам (вызовы) находятся в теле рабочей программы, а подпрограммы располагаются после конца программы (после инструкции НСТ 77). В подпрограмме не допускается применять инструкции структурирования.

ВНП

пн — номер подпрограммы

Когда цель, содержащая инструкцию ВНП становится истинной, процессор выполняет подпрограмму с соответствующим номером, после конца подпрограммы, процессор возвращается на выполнение инструкции, следующей за инструкцией ВНП.

НЧАЛО ПОДПРОГРАММЫ

ННП

пн — номер подпрограммы

Инструкция ВНП соответствует инструкции ННП с тем же номером. После ННП следует подпрограмма.

КОНЕЦ ПОДПРОГРАММЫ

НВН 77

Подпрограмма обязательно должна заканчиваться инструкцией НВН 77.

КОНЕЦ МАССИВА ПОДПРОГРАММ

ННП 77

Рабочая программа завершается инструкцией ННП 77.

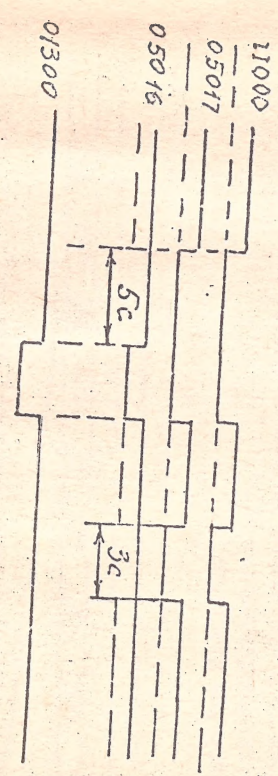
Пример программы с двумя подпрограммами:

НСТ 00
 НВН 00
 -11- 112000
 -()- 006 04
 -11- 112 01

Программа

Номер инструкции	Мнемоник	Примечание
0000	НСТ 00	
1	НВН 00	
2	-11- 110 00	
3	ТОД 050	
4	ПРК 0050	
5	-11- 050 16	
6	-()- 013 00	
7	НСТ 77	
8	ННП 77	

Временная диаграмма



НАКАЛИВАЮЩИЙ ТАЙМЕР

ТНД - таймер накаливающего с дискретностью отсчета времени Δt , $\Delta t = 0,1c$.

ТНД - с дискретностью отсчета времени Δt .

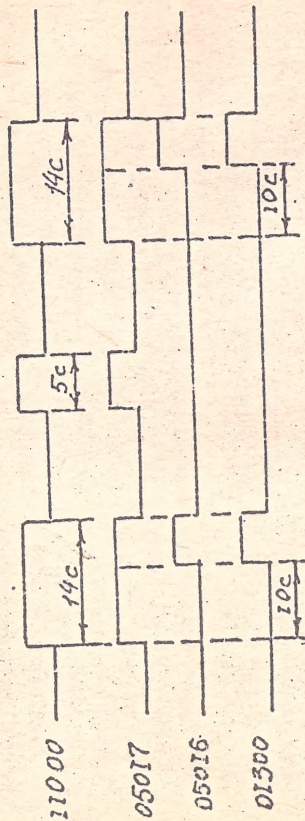
ПРК к к к к к

ТНД п п п п п

ПРС м м м м м

Накаливающий таймер считает дискретные времена, когда его управляющая цепь логически истинна, и сохраняет свое накопленное значение, если условия в цепи становятся ложными, а также при потере питания.

Временная диаграмма:



ТАЙМЕР С ЗАДЕРЖКОЙ НА ОТКЛЮЧЕНИЕ

ТОД - таймер с задержкой на отключение с дискретностью

отсчета времени 0,1 с;

ТОС - с дискретностью отсчета 1с.

ТОС *nnn* ; ТОО *nnn* ;

ПРК *kkkk* ; ПРС *mmmm* ;

ТОД *nnn* ; ТОО *nnn* ;

ПРК *kkkk* ; ПРС *mmmm* ;

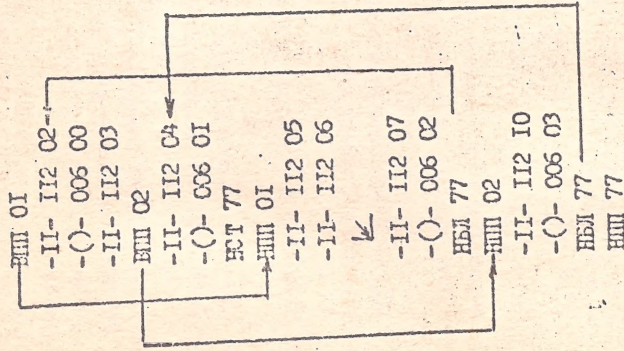
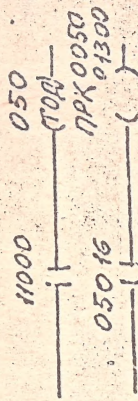
таймер с задержкой на отключение считает количество временных дискрет и хранит их счет в накопленном значении.

таймер с задержкой на отключение начинает отсчет, когда уровень в его управляющей цепи становится ложным и прекращает счет пока накопленное значение не достигнет уставки.

Бит 17 отсчитывает ложное и истинное состояние управляющей цепи. Бит 15 отображает дискрету времени. Бит 16 - завершение операции.

Пример с использованием таймера на отключение:

выход 013 00 должен отключиться через 5с, после размыкания контакта 110 00.



При включении бита 112 01 происходит взвод подпрограммы 1. Процессор репит функции: 11205, 11206 + 11207 = 00602; и по команде БИТ 77 вернется на чтение контакта 112 02.

2-8. Инструкции таймеров и счетчиков

Группа инструкций таймеров и счетчиков включает следующие инструкции:

- таймер с задержкой на включение;
- таймер с задержкой на отключение;
- накапливающий таймер;
- счетчик на вычитание;
- счетчик на сложение;
- счетчик циклов.

Благодаря этим инструкциям ПК имеет возможности аналогичные реле времени, полупроводниковым, переменным и счетным устройствам. В ПК таймеры и счетчики реализованы программно с помощью *резидентной* программного обеспечения. Тем не менее они должны программироваться пользователями, который задает условия их работы.

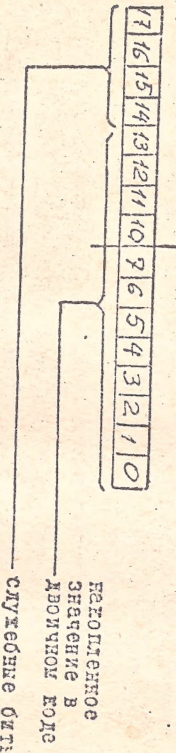
Таймеры отсчитывают интервалы времени, счетчики считают события по алгоритмам, определяют логикой программы пользователей. Каждая инструкция таймера и счетчика связана с уставкой - заданное заданным числом. Когда накопленное значение достигает заданной величины, включается бит состояния, который может быть использован для управления выходным устройством.

Накопленное значение - текущее количество безвыходных интервалов времени, которое сосчитал таймер или количество событий, сосчитанных счетчиком.

Уставка и накопленное значение для таймеров и счетчиков могут изменяться от 0 до 4095.

Уставка таймера и счетчика может задаваться в виде константы, которая в программе следует за инструкцией таймера - ПРК, или адресом слова, где находится уставка - ПРС.

По адресу, который следует за инструкцией таймера или счетчика, хранится информация о накопленном значении и служебные биты:



Дискретность счета - это базовые интервалы времени.

Для ПК доступны следующие значения базового интервала: Ic; O, Ic. Таймер, таймер с задержкой на включение и с дискретностью счета Ic - ТВ, а ТВИ этот же таймер с дискретностью счета Ic.

Если в программе используется таймер с дискретностью счета времени O, Ic, то время цикла не должно превышать 0,05 с. Для таймера с дискретностью счета Ic время цикла не должно превышать 0,5 с. Таймер считывает только до половины сегмента и активной дорожки.

ТАЙМЕР С ЗАДЕРЖКОЙ НА ВКЛЮЧЕНИЕ
ТВИ - дискретность отсчета времени O, Ic,
ТВО - дискретность отсчета времени Ic.
ТВИ nnn; ТВИ nnn; ТВО nnn; ТВО nnn;
ПРК kkk; ПРС mmm; ПРК kkk; ПРС mmm;
где nnn - слово, на котором задан таймер;
kkk - константа в десятичной форме от 0 до 4095;
mmm - слово, в котором записана константа.

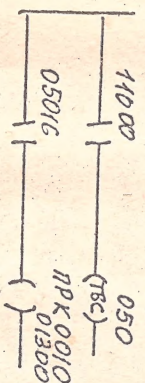
Когда условия в управляющей цепи таймера становятся истинными, таймер начинает считать с указанной дискретностью, нарабатывая свое накопленное значение на единицу в каждом интервале.

Когда накопленное значение достигает уставки, таймер останавливает счет и включает выходной сигнал (16) бит завершения операции. Этот бит остается включенным, пока условия в цепи не станут ложными. Если условия в цепи становятся ложными до окончания отсчета времени, таймер останавливается и его накопленное значение сбрасывается в ноль.

Бит семисигналы (17) включается в момент, когда условия в цепи становятся истинными и сбрасывается, когда условия в цепи становятся ложными.

Бит пятидцати (15) - отображает дискрету времени.

Пример с использованием таймера на включение:
Выход О13 00 должен включиться через 10с, после завершения контакта I10 00



Программа:

Номер инструкции	Мнемоника	Примечание
0000	НСТ 00	
1	НВИ 00	
2	-11- I10 00	
3	ТВС 050	
4	ПРК 0010	
5	-11- 050 16	
6	-(-)- О13 00	
7	НСТ 77	
8	ННН 77	